

**PROTOTIPO DE UN DISPOSITIVO QUE PERMITA VERIFICAR LA CANTIDAD
DE COMBUSTIBLE EN LOS TANQUES DE ALMACENAMIENTO DE LAS
PLANTAS ELÉCTRICAS DE LA CENTRAL TELEFÓNICA DE COLÓN (EMCALI)**

**MARCEL ENRIQUE MIDEROS HURTADO
JOSE FERNANDO TORO MORENO**

**UNIVERSIDAD AUTONOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE AUTOMATICA Y ELECTRONICA
PROGRAMA DE INGENIERIA ELECTRÓNICA
SANTIAGO DE CALI
2008**

**PROTOTIPO DE UN DISPOSITIVO QUE PERMITA VERIFICAR LA CANTIDAD
DE COMBUSTIBLE EN LOS TANQUES DE ALMACENAMIENTO DE LAS
PLANTAS ELÉCTRICAS DE LA CENTRAL TELEFÓNICA DE COLÓN (EMCALI)**

**MARCEL ENRIQUE MIDEROS HURTADO
JOSÉ FERNANDO TORO MORENO**

Pasantía para optar al título de Ingeniero Electrónico

**Director
JORGE IVAN VELANDIA ROMERO
Ingeniero Electrónico
Especialista en automatización de equipos y procesos industriales**

**UNIVERSIDAD AUTONOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE AUTOMATICA Y ELECTRONICA
PROGRAMA DE INGENIERIA ELECTRÓNICA
SANTIAGO DE CALI
2008**

Nota de aceptación:

Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Universidad Autónoma de Occidente para optar al título de Ingeniero Electrónico.

Ing. OSCAR FERNANDO AGREDO
Jurado

Santiago de Cali Enero de 2008

AGRADECIMIENTOS

A nuestras familias por su apoyo permanentemente, a nuestros compañeros y amigos por la compañía durante este largo camino, y a nuestros profesores compartir sus conocimientos y experiencias.

CONTENIDO

	Pág.
GLOSARIO	13
RESUMEN	15
INTRODUCCION	16
1. PLANTEAMIENTO DEL PROBLEMA	17
1.1. DEFINICIÓN DEL PROBLEMA	17
1.2. MARCO TEORICO	17
1.2.1. Sistemas SCADA	18
1.2.2. Medidores de nivel de liquidos	24
1.2.3. Digitalización	33
1.2.4. Microcontrolador	38
1.2.5. Transmision de datos	42
1.2.6. Familia de protocolos de internet	47
1.2.7. Lenguaje de programación	55
1.2.8. Base de datos	61
1.2.9. Redes de computadores	64
1.3. ANTECEDENTES	68
1.4. OBJETIVO GENERAL	68
1.5. OBJETIVOS ESPECIFICOS	69
1.6. JUSTIFICACION	69

1.7. METODOLOGIA	70
1.8. CRONOGRAMA	70
2. DISEÑO CONCEPTUAL	71
2.1. ESTUDIO Y ANÁLISIS	71
2.1.1. Identificación de los requerimientos del sistema	71
2.1.2. Clasificación de los requerimientos del sistema	72
2.1.3. Generación de alternativas de solución	75
2.1.4. Análisis de las alternativas de solución	78
3. DISEÑO ARQUITECTURAL	87
3.1. DIAGRAMA DE BLOQUES	87
3.2. DIAGRAMAS DE FLUJO	87
4. DISEÑO DETALLADO	92
4.1. DESARROLLO DEL SISTEMA	92
4.1.1. Descripción del sistema	92
5. RESULTADOS Y PRUEBAS	108
5.1. COSTO DEL DESARROLLO DEL PROTOTIPO	109
5.2. CONSUMO DE POTENCIA DEL PROTOTIPO	110
6. CONCLUSIONES	111
7. RECOMENDACIONES	112
BIBLIOGRAFIA	113
ANEXOS	114

LISTA DE TABLAS

	Pág.
Tabla 1. Comparación entre microcontrolador y autómeta Programable	23
Tabla 2. Fabricantes de μ C más conocidas en el mercado	42
Tabla 3. Modelo TCP/IP	48
Tabla 4. Formato de los segmentos TCP	49
Tabla 5. Cronograma de trabajo	70
Tabla 6. Clasificación de los requerimientos del sistema	72
Tabla 7. Jerarquización de los requerimientos	73
Tabla 8. Métricas y unidades de medida que determinan al dispositivo	73
Tabla 9. Metricas	74
Tabla 10. Matriz métricas y relación con los requerimientos	74
Tabla 11. Comparacion de soluciones funcion medicion	79
Tabla 12. Comparacion de soluciones funcion acondicionamiento	80
Tabla 13. Comparacion de soluciones funcion digitalizacion	81
Tabla 14. Comparacion de soluciones funcion transmision de datos	82
Tabla 15. Comparacion de soluciones funcion HMI	83
Tabla 16. Tabla morfológica	84
Tabla 17. Tabla morfológica Selección de la solución	84
Tabla 18. Tabla combinación de soluciones	84

Tabla 19. Comparación precios	96
Tabla 20. Velocidades de transmisión serial Atmel89c52	96
Tabla 21. Selección de la señal de entrada	106
Tabla 22. Gastos estimados para la realización del proyecto	109
Tabla 23. Costo de los elementos del prototipo	110
Tabla 24. Consumo de Potencia del dispositivo	110

LISTA DE FIGURAS

	Pág.
Figura 1. Tanque abierto	24
Figura 2. Tanque cerrado	24
Figura 3. Medidor de cristal	25
Figura 4. Medidor de flotador	26
Figura 5. Medidor de tipo burbujeo	28
Figura 6. Medidor de presión diferencial de diafragma	29
Figura 7. Medidor de nivel capacitivo	30
Figura 8. Medidor de nivel ultrasónico	31
Figura 9. Sistema de medición de nivel radiactivo	32
Figura 10. Esquema general ADC	35
Figura 11. Diagrama del ADC0808	36
Figura 12. Transmisión en paralelo	43
Figura 13. Transmisión en serie	43
Figura 14. Formato de un carácter	44
Figura 15. Establecimiento de la conexión	50
Figura 16. Cierre de una conexión según el estándar	51
Figura 17. Medidor de sonda EMCALI	68
Figura 18. Caja negra del sistema	76

Figura 19. Descomposición funcional del sistema	76
Figura 20. Función HMI	78
Figura 21. Diagrama de bloques del sistema	87
Figura 22. Diagramas de flujo	87
Figura 23. Diagrama de la conversión análoga digital	88
Figura 24. Diagrama transmisión serial	88
Figura 25. Diagrama de interfaz grafica (HMI)	89
Figura 26. Diagrama de configuración inicial	89
Figura 27. Diagrama de procesos recepción y envío de datos	90
Figura 28. Diagrama conectarse a otras centrales	90
Figura 29. Diagrama especificar niveles de referencia	91
Figura 30. Diagrama visualización de datos y alarmas	91
Figura 31. Acondicionamiento de la señal del sensor de presión	93
Figura 32. ADC0808	94
Figura 33. Configuración LM555	95
Figura 34. Dispositivo acondicionamiento y comunicación serial	97
Figura 35. Microcontrolador Atmel89c52	97
Figura 36. Circuito Reset microcontrolador	98
Figura 37. Ventana configuración Inicial	98
Figura 38. Base de datos	99
Figura 39. Ventana dirección IP	99

Figura 40. Ventana principal	100
Figura 41. Menú de configuración	100
Figura 42. Ventana niveles de referencia	101
Figura 43. Ventana niveles de referencia central1	101
Figura 44. Ventana niveles de referencia central2	101
Figura 45. Ventana niveles de referencia central3	102
Figura 46. Ventana monitoreo central1	102
Figura 47. Ventana monitoreo Central1 Tanque1	103
Figura 48. Ventana monitoreo Central2	103
Figura 49. Ventana monitoreo Central2 Tanque1	103
Figura 50. Ventana de alerta	104
Figura 51. Ventana de conexión Rechazada	104
Figura 52. Diseño esquemático del sistema	105
Figura 53. Diseño impreso del sistema	105
Figura 54. Ampliación de la capacidad de tanques registrados	106
Figura 55. Diseño del prototipo del dispositivo de comunicación	107
Figura 56. Simulación del dispositivo de comunicación serial	108
Figura 57. Instalación de sistema de monitoreo	109

LISTA DE ANEXOS

	Pág.
Anexo A. Manual de usuario	114
Anexo B. Características microcontrolador atmel89c52	118
Anexo C. Características sensor de presión diferencial.	120
Anexo D. Características LM555	122
Anexo E. Hoja de seguridad ACPM	124
Anexo F. Programación y configuración del microcontrolador	126
Anexo G. Código fuente de la configuración inicial	130
Anexo H. Código fuente de la direcciones IP	134
Anexo I. Código fuente de la niveles de referencia	137
Anexo J. Código fuente principal	143
Anexo K. Código fuente del nivel de cada Tanque	155

GLOSARIO

ACPM: aceite combustible para motores Diesel.

BASE DE DATOS: conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente.

CISC: computadoras con un conjunto de instrucciones complejo.

DATAGRAMA: unidad o conjunto de datos para la transmisión de información.

DCE: equipo de comunicación de datos.

DLL: librerías dinámicas

DTE: equipo terminal de datos.

ETHERNET: tecnología de redes de computadoras de área local basada en tramas de datos.

HMI: una interfaz hombre es el aparato que presenta los datos a un operador (humano) y a través del cual éste controla el proceso.

IDE: entorno de desarrollo integrado.

IP: protocolo de Internet.

JAVA: lenguaje de programación de alto nivel orientado a objetos

MTU: unidad máxima de transferencia.

RISC: computadoras con un conjunto de instrucciones reducido.

RS-232: interfaz que designa para el intercambio serie de datos binarios entre un DCE y un DTE

RTU: unidad de Terminal Remota.

SCADA: sistema de control supervisión y adquisición de datos.

SOCKETS: concepto abstracto por el cual dos programas pueden intercambiarse cualquier flujo de datos, esta definido por una dirección IP, un protocolo y un

número de puerto.

TCP: protocolo de control de transmisión.

UART: transmisor y receptor síncrono/asíncrono universal.

UDP: protocolo para el intercambio de datagramas.

VISUALBASIC: lenguaje de programación de alto nivel con interfaz grafica basado en basic.

RESUMEN

En un sistema SCADA para el monitoreo de señales análogas o digitales tomadas de un sensor en este caso un medidor de nivel se utiliza una unidad que procese las señales y las convierta en información para su visualización en este proyecto se usara un microcontrolador con las siguientes funciones: muestreo, conversión A/D, codificación de la información, multiplexación de las señales y comunicación con el PC. Para la comunicación entre el PC y el microcontrolador se usara una interfaz de comunicación serial, la interfaz mas común es la interfaz RS-232 diseñada para distancias cortas. En este sistema no se necesitaran altas velocidades de comunicación entre el microcontrolador y el PC porque se monitoreara un tanque, caracterizado por ser un sistema dinámico con un tiempo de respuesta lento.

Para realizar la comunicación entre centrales se usara un programa con arquitectura cliente servidor donde se visualice, registre y transmita la información recolectada de los sensores ubicado en cada uno de los tanques de la central principal y de las centrales secundarias. El uso de la arquitectura cliente servidor permitira el intercambio de información entre dos o mas dispositivos conectados al servidor ubicado en la central principal, en este sistema tanto el servidor como el cliente están conectados por medio de una red con una velocidad promedio de 10Mbps, la aplicación encargada de los procesos de visualización, recolección de información y conexión directa entre cliente y servidor se desarrollara en el entorno de programación Visual Basic.

INTRODUCCION

La central telefónica de colon esta interesada en modernizar sus sistemas de control, en este caso el monitoreo de la cantidad de combustible que hay en los tanque de almacenamiento de sus plantas eléctricas de emergencia.

Esta modernizacion se pretende llevar a cabo mediante la implementación de un sistema SCADA que permita obtener una lectura exacta de la cantidad de combustible que hay en los tanques de las Plantas Eléctricas en tiempo real, garantizando, asi la prevencion de cualquier falla en el correcto funcionamiento del suministro de energia de la central. En este proyecto se desarrollara en detalle cada uno de los elementos necesarios para realizar el monitoreo en forma centralizada de los tanques de una de las centrales y su ampliación para el seguimiento, del estado de los tanques de suministro de otras centrales por medio de la red de area extendida que existe entre la central principal en este caso la central de colon y las otras centrales secundarias.

El desarrollo se levara a cabo siguiendo los liniamientos y características de un sistema SCADA, entre ellas la continua interaccion con el usuario, el registro de cualquier evento fuera de lo normal, y el historial del comportamiento del sistema monitoreado en este caso, los tanques de combustible de una central.

1. PLANTEAMIENTO DEL PROBLEMA

1.1. DEFINICIÓN DEL PROBLEMA

En La Central Telefónica de Colon cuentan con equipos de apoyo los cuales son; Aires Acondicionados, Plantas Rectificadoras, Moto Bomba, Banco de Baterías y Plantas Eléctricas; estas ultimas son utilizadas en caso de un corte en el fluido eléctrico, este respaldo debe de ser de forma confiable y seguro al darse el corte del suministro de energía, ya que los equipos de conmutación y aires acondicionados, no pueden dejar de funcionar ya que de hacerlo el sistema de comunicación telefónica colapsaría, dejando incomunicado a los usuarios de EMCALI.

Por esta razón se busca que los equipos de apoyo se encuentren en un muy buen estado y en óptimas condiciones en el momento que se requieran.

En estos momentos las Plantas Eléctricas no cuentan con un sistema centralizado, que nos permita verificar en tiempo real la cantidad de combustible que hay en su tanque de almacenamiento, esta medición se hace de forma visual ya que cuenta con un sistema de medición visual, pero debido al tipo de combustible (ACPM) que utilizan las Plantas Eléctricas, la visibilidad se hace difícil y en algunos casos es imposible ver el nivel en el que se encuentra el tanque, dejando como única opción la utilización de un pedazo de madera (un palo) para su medición, haciendo que el procedimiento sea un poco rudimentario por no decir que obsoleto. Además de esto el inconveniente más grande es que no es solo una Planta Eléctrica a la que hay que hacerle este control si no que son alrededor de cincuenta y cinco (55) a las que hay que hacerles este control día a día, lo cual demanda tiempo y gastos económicos en el traslado de las personas que se encargan de verificar el nivel de combustible y en el reabastecimiento del mismo.

1.2. MARCO TEORICO

Este proyecto está enmarcado en el área de la instrumentación, electrónica digital y programación. El proyecto se desarrollara aplicando los conocimientos adquiridos en lenguajes de programación, programación de microcontroladores, manejo de redes y acondicionamiento de señales análogas y digitales. El desarrollo de este proyecto se encuentra incluido en el ámbito de los sistemas SCADA que se basan en el control, monitoreo y adquisición de datos de un proceso o planta industrial.

1.2.1 Sistemas SCADA. La información recopilada con los sistemas SCADA se utiliza para realizar una serie de análisis o estudios con los que se pueden obtener valiosos indicadores que permitan una retroalimentación_sobre un operador o sobre el propio proceso, tales como:

- Indicadores sin retroalimentación inherente (no afectan al proceso, sólo al operador):
 - Estado actual del proceso: Valores instantáneos.
 - Desviación o deriva del proceso: Evolución histórica y acumulada.
- Indicadores con retroalimentación inherente (afectan al proceso, después al operador):
 - Generación de alarmas:
 - HMI Human Machine Interface (Interfaces hombre-máquina);
 - Toma de decisiones:
 - Mediante operatoria humana:
 - Automática (mediante la utilización de sistemas basados en el conocimiento sistemas expertos.

Por medio de un sistema central que monitorea y controla un sitio completo. La mayor parte del control del sitio es en realidad realizada automáticamente por una Unidad Terminal Remota (RTU) o por un Controlador Lógico Programable (PLC).

Las funciones de control del servidor están casi siempre restringidas a reajustes básicos del sitio o capacidades de nivel de supervisión. Por ejemplo un PLC puede controlar el flujo de agua fría a través de un proceso, pero un sistema SCADA puede permitirle a un operador cambiar el punto de consigna (set point) de control para el flujo, y permitirá grabar y mostrar cualquier condición de alarma como la pérdida de un flujo o una alta temperatura. La realimentación del lazo de control es cerrada a través del RTU o el PLC; el sistema SCADA monitorea el desempeño general de dicho lazo.

Un sistema SCADA incluye un hardware de señal de entrada y salida, controladores, interfaz hombre-máquina, redes, comunicaciones, base de datos y software.

► Interfaz Hombre – Maquina. La industria de HMI nació esencialmente de la necesidad de estandarizar la manera de monitorear y de controlar múltiples sistemas remotos, PLCs y otros mecanismos de control. Aunque un PLC realiza automáticamente un control pre-programado sobre un proceso, normalmente se distribuyen a lo largo de toda la planta, haciendo difícil recoger los datos de manera manual, los sistemas SCADA lo hacen de manera automática.

Un HMI puede tener también vínculos con una base de datos para proporcionar las tendencias, los datos de diagnóstico y manejo de la información así como un cronograma de procedimientos de mantenimiento, información logística, esquemas detallados para un sensor o máquina en particular, incluso sistemas expertos con guía de resolución de problemas.

Una adecuada interfaz hombre maquina busca cumplir los siguientes requisitos:

- Asegura que el observador comprenda la situación representada.
- Crear condiciones para la toma de decisiones correctas.
- Que los equipos se utilicen en forma óptima y segura.
- Garantizar confiabilidad al máximo.
- Cambiar con facilidad los niveles de actividades del operador.

Los aspectos esenciales que se realizan en la comunicación hombre- maquina son:

- Indicar el estado del proceso
- Indicar situaciones de alarma.
- Ejecución de acciones de mando.

► Indicaciones del Estado del Proceso. Para esto se pueden utilizar: Equipos convencionales, terminales de video, impresoras, registradores, diodos emisores de luz (LED). Para la selección en pantalla se pueden emplear: Mouse, teclado, lápiz óptico, “touch screen”, etc.

► Indicaciones del las Alarmas. Una de las partes más importantes de la implementación de SCADA son las alarmas. Una alarma es un punto de estado digital que tiene cada valor NORMAL o ALARMA. La alarma se puede crear en cada paso que los requerimientos lo necesiten. Un ejemplo de un alarma es la luz de "tanque de combustible vacío" del automóvil. El operador de SCADA pone atención a la parte del sistema que lo requiera, por la alarma. Pueden enviarse por correo electrónico o mensajes de texto con la activación de una alarma, alertando

al administrador o incluso al operador de SCADA. Busca informar al operador de una situación anormal.

► Las alarmas se pueden representar:

- En la pantalla, mediante símbolos que aparecen intermitentemente, cambios repetidos de color en el nombre de alguna variable o grupo, intermitencia de textos, mensajes, etc.
- Mediante el uso de videos o animaciones específicos.
- Mediante indicación sonora.
- Utilizando impresoras para reportar los mensajes de alarma.

► Ejecución de Acciones de Mando. Se pueden realizar por técnicas convencionales (pulsadores, interruptores, potenciómetros, etc.), o mediante teclados, lápiz óptico, mouse, pantallas táctiles, etc.

○ Características del Puesto de Mando.

- Comodidad para el operador.
- Tener presente la cantidad de operadores y horas de trabajo en dependencia de la intensidad y responsabilidad de la industria.
- La operación de los mandos de los mandos debe ser fácil y segura. Debe existir buena visibilidad en todo el cuarto de mando.

► Soluciones de Hardware. La solución de SCADA a menudo tiene componentes de sistemas de control distribuido, DCS (Distributed Control System). El uso de RTUs o PLCs sin involucrar computadoras maestras está aumentando, los cuales son autónomos ejecutando procesos de lógica simple. Frecuentemente se usa un lenguaje de programación funcional para crear programas que corran en estos RTUs y PLCs, siempre siguiendo los estándares de la norma IEC 61131-3. La complejidad y la naturaleza de este tipo de programación hace que los programadores necesiten cierta especialización y conocimiento sobre los actuadores que van a programar.

Aunque la programación de estos elementos es ligeramente distinta a la programación tradicional, también se usan lenguajes que establecen procedimientos, como pueden ser FORTRAN, C o Ada95. Esto les permite a los ingenieros de sistemas SCADA implementar programas para ser ejecutados en RTUs o un PLCs.

► Componentes del Sistema. Los tres componentes de un sistema SCADA son:

- Múltiples Unidades de Terminal Remota (también conocida como RTU o Estaciones Externas).
- Estación Maestra y Computador con HMI.
- Infraestructura de Comunicación

► Unidad de Terminal Remota (RTU). La RTU se conecta al equipo físicamente y lee los datos de estado como los estados abierto/cerrado desde una válvula o un intercambiador, lee las medidas como presión, flujo, voltaje o corriente. Por el equipo el RTU puede enviar señales que pueden controlarlo: abrirlo, cerrarlo, intercambiar la válvula o configurar la velocidad de la bomba. La RTU puede leer el estado de los datos digitales o medidas de datos analógicos y envía comandos digitales de salida o puntos de ajuste analógicos.

► Estación Maestra. El termino "Estación Maestra" se refiere a los servidores y el software responsable para comunicarse con el equipo del campo (RTUs, PLCs, etc) en estos se encuentra el software HMI corriendo para las estaciones de trabajo en el cuarto de control, o en cualquier otro lado. En un sistema SCADA pequeño, la estación maestra puede estar en un solo computador, A gran escala, en los sistemas SCADA la estación maestra puede incluir muchos servidores, aplicaciones de software distribuido, y sitios de recuperación de desastres.

El sistema SCADA usualmente presenta la información al personal operativo de manera gráfica, en la forma de un diagrama de representación. Esto significa que el operador puede ver un esquema que representa la planta que está siendo controlada. Por ejemplo un dibujo de una bomba conectada a la tubería puede mostrar al operador cuanto fluido esta siendo bombeado desde la bomba a través de la tubería en un momento dado. El operador puede cambiar el estado de la bomba a apagado. El software HMI mostrará el promedio de fluido en la tubería incrementándose en tiempo real. Los diagramas de representación pueden consistir en gráficos de líneas y símbolos esquemáticos para representar los elementos del proceso, o pueden consistir en fotografías digitales de los equipos sobre los cuales se animan las secuencias.

El paquete HMI para el sistema SCADA típicamente incluye un programa de dibujo con el cual los operadores o el personal de mantenimiento del sistema usan para cambiar la manera que estos puntos son representados en la interfase. Estas representaciones puede ser tan simples como por ejemplo luces de tráfico en pantalla, los cuales representan el estado actual de trafico de un, o tan complejo

como una pantalla de multiproyector representando posiciones de todos los elevadores en un rascacielos o todos los trenes de una vía férrea.

- Infraestructura y Métodos de Comunicación. Los sistemas SCADA tienen tradicionalmente una combinación de radios y señales directas seriales o conexiones de modem para conocer los requerimientos de comunicaciones, incluso Ethernet e IP sobre SONET es también frecuentemente usada en sitios muy grandes como ferrocarriles y estaciones de poder (energía eléctrica).

- SCADA en Áreas Industriales:

- Monitorizar procesos químicos, físicos o de transporte en sistemas de suministro de agua, para controlar la generación y distribución de energía eléctrica, de gas o en oleoductos y otros procesos de distribución.

- Gestión de la producción (facilita la programación de la fabricación)

- Mantenimiento (proporciona magnitudes de interés tales para evaluar y determinar modos de fallo, MTBF, índices de Fiabilidad, entre otros)

- Control de Calidad (proporciona de manera automatizada los datos necesarios para calcular índices de estabilidad de la producción CP y CPk, tolerancias, índice de piezas NOK/OK, etc

- Administración (actualmente pueden enlazarse estos datos del SCADA con un servidor ERP (Enterprise Resource Planning o sistema de planificación de recursos empresariales), e integrarse como un módulo más)

- Tratamiento histórico de información (mediante su incorporación en bases de datos).

- Comparación entre las opciones mas comunes de monitoreo y control en el desarrollo de un sistema scada.

Tabla 1. Comparación entre microcontrolador y Autómata programable

Microcontrolador	Autómata Programable
La mayoría de los microcontroladores pueden ejecutar entre 1 millón y 5 millones de instrucciones por segundo. Es ideal para procesamiento muy rápidos en tiempo real.	Un autómata programable ejecuta entre 300 y 30.000 instrucciones por segundo.
Se suele programar con un lenguaje de bajo nivel tipo ensamblador o C o bien alto nivel como el BASCOM LT, donde el usuario debe desarrollarse manejadores para los diferentes periféricos.	Se programa siempre mediante un lenguaje de alto nivel, el BASIC, en el que todos los manejadores de periféricos ya están preprogramados. Con él se pueden desarrollar aplicaciones en menor tiempo.
La utilización de un microcontrolador requiere de un periodo de aprendizaje relativamente largo.	Una persona no experimentada o incluso un niño puede comenzar a realizar programas el primer día. Apenas requiere de tiempo de aprendizaje o conocimientos previos.
Es mucho más flexible en el sentido de que todas las entradas/salidas son de propósito general. Esta flexibilidad obliga a su usuario a desarrollar su propia electrónica.	Cada entrada/salida tiene una función preasignada, esto es, LCD, teclado, relés, triacs, optoacopladores, ... Toda la electrónica básica ya está desarrollada.
Su electrónica es mucho más simple, por lo que su consumo es inferior a 50 mA. Se pueden lograr aplicaciones concretas con consumos muy bajos del orden de los 5 mA.	Su consumo, y peso, es relativamente alto, no siendo adecuado para aplicaciones portátiles o de larga duración que deban alimentarse a pilas. Su consumo es superior a los 500 mA, pudiendo un sistema con 64 entradas/salidas llegar a los 3A.
Para dejar los programas residentes requiere de un programador de EPROMs externo.	No requiere de ningún dispositivo de grabación externo, todos los datos son almacenados permanentemente a través del puerto serie.
A mismo número de entradas/salidas tiene un precio notablemente inferior, pero debe tenerse en cuenta que trabaja con tensiones de 5 voltios y corrientes muy pequeñas del orden de miliamperios. Sus señales están disponibles en conectores de 1,27 milímetros.	Su precio es superior, pero las entradas/salidas pueden aceptar directamente tensiones de 5, 12, 24, 110 y 220 voltios. Las señales están disponibles en regletas de paso 5 milímetros.

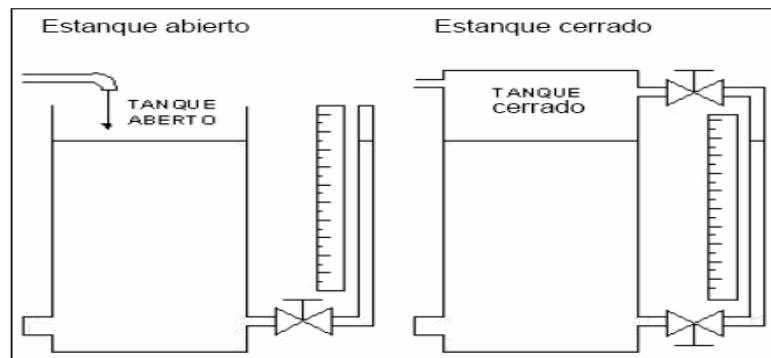
1.2.2. Medidores de nivel de líquidos. Los medidores de nivel de líquidos trabajan midiendo; directamente la altura de líquido sobre una línea de referencia, la presión hidrostática, el desplazamiento producido en un flotador por el propio líquido contenido en el tanque del proceso, o aprovechando características eléctricas del líquido.

Los Instrumentos de Medida Directa se dividen en:

- Medidor de sonda.
- Medidor de cinta y plomada.
- Medidor de nivel de cristal.
- Medidor de flotador.

► Medidor de Sonda. Consiste en una varilla o regla graduada, de la longitud conveniente para introducirla dentro del depósito. La determinación del nivel se efectúa por la lectura directa de la longitud mojada por el líquido. En el momento de la lectura el tanque debe estar abierto a presión atmosférica. Se utiliza generalmente en tanques de gasolina (Ver Figura 1). Otro medidor consiste en una varilla graduada, con un gancho que se sumerge en el seno del líquido y se levanta después hasta que el gancho rompe la superficie del líquido

Figura 1. Tanque abierto, Figura 2. Tanque cerrado

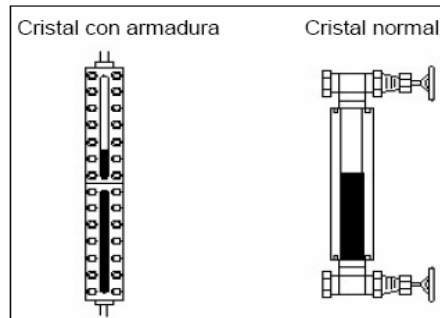


Fuente: BELTRÁN PROVOSTE, Cristhian. Medición de nivel [en línea]. Barcelona: infoPLC.net, 2008. Modificada por última vez 13 septiembre 2007. [Consultado 03 Marzo de 2007]. Disponible Internet: [http://www.infoplc.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.p df](http://www.infoplc.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.pdf)

► Medidor de Cinta y Plomada. Este sistema es parecido a los anteriores, consta de una cinta graduada y un plomo en la punta. Se emplea cuando es difícil que la regla tenga acceso al fondo del tanque.

► Medidor de Cristal. Consiste en un tubo de vidrio con sus extremos conectados a bloques metálicos y cerrados por prensaestopas que están unidos al tanque generalmente mediante tres válvulas, dos de cierre de seguridad en los extremos del tubo para impedir el escape del líquido en caso de rotura del cristal y una de purga (Ver Figura 3). El nivel de cristal normal se emplea para presiones hasta 7 bar. A presiones más elevadas el cristal es grueso, de sección rectangular y está protegido por una armadura metálica.

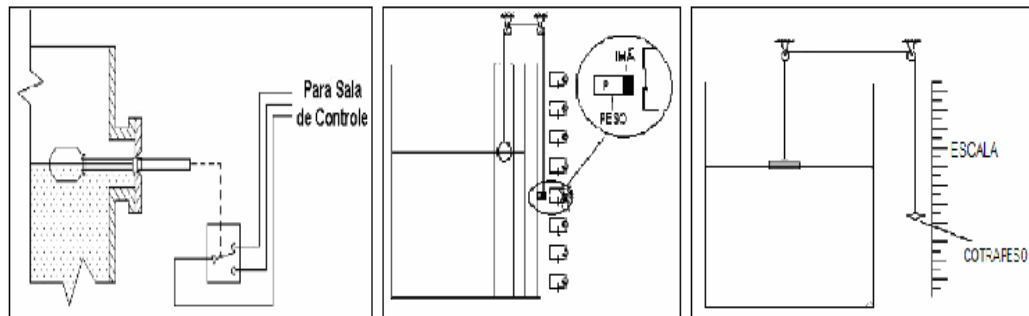
Figura 3. Medidor de cristal



Fuente: BELTRÁN PROVOSTE, Cristhian. Medición de nivel [en línea]. Barcelona: infoPLC.net, 2008. Modificada por última vez 13 septiembre 2007. [Consultado 03 Marzo de 2007]. Disponible Internet: http://www.infoplc.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.pdf

► Medidor de Flotador. Consiste en un flotador ubicado en el seno del líquido y conectado al exterior del tanque indicando directamente el nivel sobre una escala graduada. Es el modelo más antiguo y el más utilizado en tanques de gran capacidad tales como los de petróleo y gasolina. Tiene el inconveniente de que las partes móviles están expuestas al fluido y pueden romperse, además el flotador debe mantenerse limpio (Ver Figura 4).

Figura 4. Medidor de flotador



Fuente: BELTRÁN PROVOSTE, Cristhian. Medición de nivel [en línea]. Barcelona: infoPLC.net, 2008. Modificada por última vez 13 septiembre 2007. [Consultado 03 Marzo de 2007]. Disponible Internet: http://www.infoplcn.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.pdf

Hay que señalar que en estos instrumentos, el flotador puede tener formas muy variadas y estar formados por materiales muy diversos según sea el tipo de fluido. Los instrumentos de flotador tienen una precisión de $\pm 0,5\%$. Son adecuados en la medida de niveles en tanques abiertos y cerrados a presión o a vacío, y son independientes del peso específico del líquido. Por otro lado, el flotador puede ajustarse en el tubo guía por un eventual depósito de los sólidos o cristales que el líquido pueda contener y además los tubos guía muy largos pueden dañarse ante olas bruscas en la superficie del líquido o ante la caída violenta del líquido en el tanque.

Los Aparatos que Miden el nivel aprovechando la presión hidrostática se dividen en:

- Medidor manométrico.
- Medidor de membrana.
- Medidor de tipo burbujeo.
- Medidor de presión diferencial de diafragma.

► Medidor Manométrico. Consiste en un manómetro conectado directamente a la inferior del tanque. El manómetro mide la presión debida a la altura de líquido h que existe entre el nivel del tanque y el eje del instrumento.

El rango de medida del instrumento corresponderá a:

$$0 - (h \cdot @ \cdot g) \text{ donde}$$

h = altura de líquido en m

@ = densidad del líquido en Kg/m³

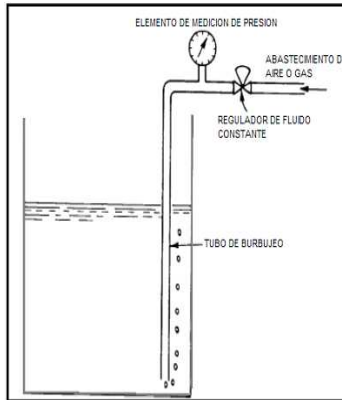
g = 9,8 m/s²

Como las alturas son limitadas, el rango de medida es bastante pequeño, de modo que el manómetro utilizado tiene un elemento de medida del tipo fuelle. El instrumento sólo sirve para fluidos limpios ya que si el líquido es corrosivo, coagula o bien tiene sólidos en suspensión, el fuelle puede destruirse o bien bloquearse perdiendo su elasticidad; por otra parte, como el rango de medida es pequeño no es posible utilizar sellos de diafragma. La medida está limitada a tanques abiertos y el nivel viene influido por las variaciones de densidad del líquido.

► **Medidor de Membrana.** Utiliza una membrana conectada con un tubo estanco al instrumento receptor. La fuerza ejercida por la columna de líquido sobre el área de la membrana comprime el aire interno a una presión igual a la ejercida por la columna de líquido. El instrumento es delicado ya que cualquier pequeña fuga del aire contenido en el diafragma destruiría la calibración del instrumento.

► **Medidor de tipo Burbujeo.** Mediante un regulador de caudal se hace pasar por un tubo (sumergido en el depósito hasta el nivel mínimo), un pequeño caudal de aire o gas inerte hasta producir una corriente continua de burbujas. La presión requerida para producir el flujo continuo de burbujas es una medida de la columna de líquido. Este sistema a pesar de ser muy ventajoso en aplicaciones con líquidos corrosivos con materiales en suspensión, necesita continuo mantenimiento (el fluido no penetra en el medidor, ni en la tubería de conexión), (Ver Figura 5).

Figura 5. Medidor de tipo burbujeo



Fuente: BELTRÁN PROVOSTE, Cristhian. Medición de nivel [en línea]. Barcelona: infoPLC.net, 2008. Modificada por última vez 13 septiembre 2007. [Consultado 03 Marzo de 2007]. Disponible Internet: http://www.infopl.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.pdf

► Medidor de Presión Diferencial de Diafragma. Consiste en un diafragma en contacto con el líquido del tanque, que mide la presión hidrostática en un punto del fondo del tanque. En un tanque abierto esta presión es proporcional a la altura del líquido en ese punto y a su peso específico, es decir:

$$P = h \cdot \rho \cdot g$$

Donde:

P = presión

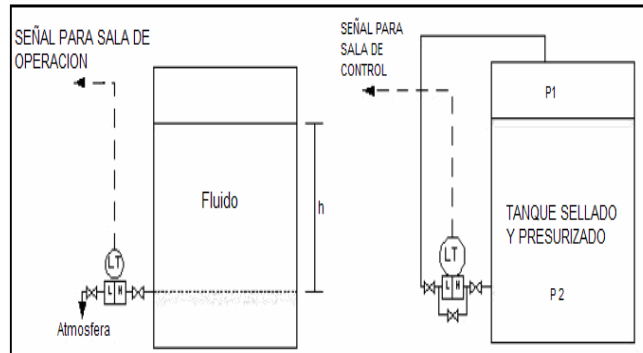
h = altura del líquido sobre el instrumento

ρ = densidad del líquido

$g = 9,8 \text{ m/s}^2$

El diafragma forma parte de un transmisor neumático, electrónico o digital de presión diferencial. Es el tipo más utilizado, en la medida de nivel de fluidos, tales como pasta de papel y líquidos con sólidos en suspensión, gasolina y petróleo, pudiendo incluso ser de montaje saliente para que el diafragma enrase completamente con las paredes interiores del tanque tal como ocurre en el caso de líquidos extremadamente viscosos en que no puede admitirse ningún recodo (Ver Figura 6).

Figura 6. Medidor de presión diferencial de diafragma



Fuente: BELTRÁN PROVOSTE, Crithian. Medición de nivel [en línea]. Barcelona: infoPLC.net, 2008. Modificada por última vez 13 septiembre 2007. [Consultado 03 Marzo de 2007]. Disponible Internet: [http://www.infoplcn.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.p](http://www.infoplcn.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.pdf)
[df](http://www.infoplcn.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.p)

La precisión de los instrumentos de presión diferencial es de $\pm 0,5\%$ en los neumáticos, $\pm 0,2\%$ a $\pm 0,3\%$ en los electrónicos, y de $\pm 0,15\%$ en los “inteligentes” con señales de salida de 4-20 ma c.c. Hay que señalar que el material del diafragma debe ser el adecuado para resistir la corrosión del fluido (existen materiales de acero inoxidable 316, monel, tantalio, hastelloy B, inoxidable recubierto de teflón).

Los instrumentos que utilizan características eléctricas del líquido para medir el nivel se clasifican en:

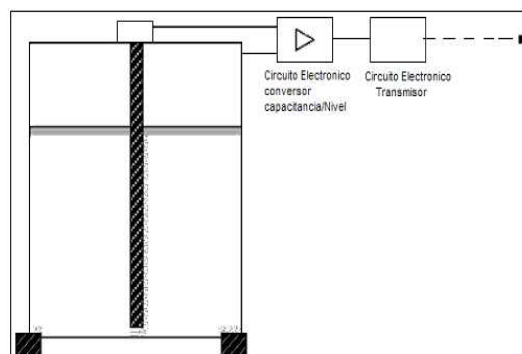
- Medidor de Nivel Conductivo o Resistivo.
- Medidor de Nivel Capacitivo.
- Medidor Ultrasónico.
- Medidor de Radiación.
- Medidor Láser.

► Medidor de Nivel Conductivo o Resistivo. Consiste en uno o varios electrodos y un rele eléctrico o electrónico que es excitado cuando el líquido moja a dichos electrodos. El líquido debe ser lo suficientemente conductor como para excitar el circuito electrónico, y de este modo el aparato puede discriminar la separación entre el líquido y su vapor, tal como ocurre, por ejemplo, en el nivel de

agua de una caldera de vapor. La impedancia mínima es del orden de los 20 MΩ/cm, y la tensión de alimentación es alterna para evitar fenómenos de oxidación en las sondas por causa del fenómeno de la electrólisis. Cuando el líquido moja los electrodos se cierra el circuito electrónico y circula una corriente segura del orden de los 2 mA; el relé electrónico dispone de un temporizador de retardo que impide su enclavamiento ante una ola del nivel del líquido o ante cualquier perturbación momentánea o bien en su lugar se disponen dos electrodos poco separados enclavados eléctricamente en el circuito. El instrumento se emplea como alarma o control de nivel alto y bajo, utiliza relees eléctricos para líquidos con buena conductividad y relees electrónicos para líquidos con baja conductividad. Montado en grupos verticales de 24 o más electrodos, puede complementar los típicos niveles de vidrio de las calderas, y se presta a la transmisión del nivel a la sala de control y a la adición de las alarmas correspondientes. El instrumento es versátil, sin partes móviles, su campo de medida es grande con la limitación física de la longitud de los electrodos. El líquido contenido en el tanque debe tener un mínimo de conductividad y si su naturaleza lo exige, la corriente debe ser baja para evitar la deterioración del producto. Por otro lado, conviene que la sensibilidad del aparato sea ajustable para detectar la presencia de espuma en caso necesario.

- Medidor de Nivel Capacitivo. Mide la capacidad del condensador formado por el electrodo sumergido en el líquido y las paredes del tanque (VerFigura 7).

Figura 7. Medidor de nivel capacitivo



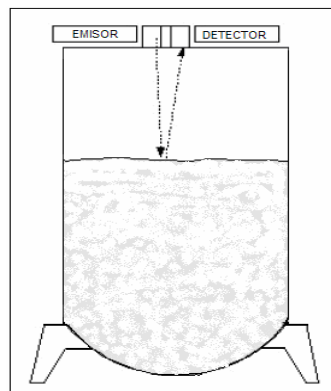
Fuente: BELTRÁN PROVOSTE, Cristhian. Medición de nivel [en línea]. Barcelona: infoPLC.net, 2008. Modificada por última vez 13 septiembre 2007. [Consultado 03 Marzo de 2007]. Disponible Internet: http://www.infoplc.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.pdf

La capacidad del conjunto depende linealmente del nivel del líquido. En fluidos no conductores se emplea un electrodo normal y la capacidad total del sistema se compone de la del líquido, la del gas superior y la de las conexiones superiores. En fluidos conductores el electrodo está aislado usualmente con teflón interviniendo las capacidades adicionales entre el material aislante y el electrodo en la zona del líquido y del gas. La precisión de los transductores de capacidad es de ± 1 %. Se caracterizan por no tener partes móviles, son ligeros, presentan una buena resistencia a la corrosión y son de fácil limpieza. Su campo de medida es prácticamente ilimitado.

Tiene el inconveniente de que la temperatura puede afectar las constantes dieléctricas (0,1 % de aumento de la constante dieléctrica / $^{\circ}\text{C}$) y de que los posibles contaminantes contenidos en el líquido puedan adherirse al electrodo variando su capacidad y falseando la lectura, en particular en el caso de líquidos conductores.

► **Medidor de Nivel Ultrasónico.** Se basa en la emisión de un impulso ultrasónico a una superficie reflectante y la recepción del eco del mismo en un receptor (Ver Figura 8). El retardo en la captación del eco depende del nivel del tanque. Los sensores trabajan a una frecuencia de unos 20 KHz. Estas ondas atraviesan con cierto amortiguamiento o reflexión el medio ambiente de gases o vapores y se reflejan en la superficie del sólido o del líquido.

Figura 8. Medidor de nivel ultrasónico

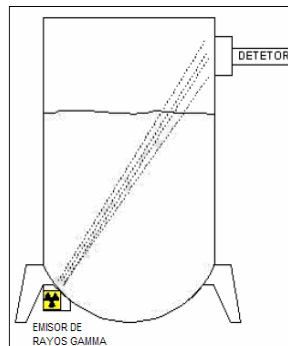


Fuente: BELTRÁN PROVOSTE, Cristhian. Medición de nivel [en línea]. Barcelona: infoPLC.net, 2008. Modificada por última vez 13 septiembre 2007. [Consultado 03 Marzo de 2007]. Disponible Internet: http://www.infoplc.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.pdf

La precisión de estos instrumentos es de ± 1 a ± 3 %. Son adecuados para todos los tipos de tanques y de líquidos o fangos pudiendo construirse a prueba de explosión. Presentan el inconveniente de ser sensibles a la densidad de los fluidos y de dar señales erróneas cuando la superficie del nivel del líquido no es nítida como es el caso de un líquido que forme espuma, ya que se producen falsos ecos de los ultrasonidos. La utilización de la computadora permite, a través de un programa, almacenar el perfil ultrasónico del nivel, y así tener en cuenta las características particulares de la superficie del líquido, tal como la espuma, con lo cual se mejora la precisión de la medida.

► El Sistema de Medición de Nivel Radiactivo. Consiste en un emisor de rayos gamma montado verticalmente en un lado del tanque y con un contador que transforma la radiación gamma recibida en una señal eléctrica de corriente continua (Ver Figura 9). Como la transmisión de los rayos es inversamente proporcional a la masa del líquido en el tanque, la radiación captada por el receptor es inversamente proporcional al nivel del líquido ya que el material absorbe parte de la energía emitida. Los rayos emitidos por la fuente son similares a los rayos X, pero de longitud de onda más corta. La fuente radiactiva pierde igualmente su radiactividad en función exponencial del tiempo. La vida media (es decir, el tiempo necesario para que el emisor pierda la mitad de su actividad) varía según la fuente empleada. En el cobalto 60 es de 5,5 años y en el cesio 137 es de 33 años y en el americio 241 es de 458 años.

Figura 9. Sistema de medición de nivel radiactivo.



Fuente: BELTRÁN PROVOSTE, Cristhian. Medición de nivel [en línea]. Barcelona: infoPLC.net, 2008. Modificada por última vez 13 septiembre 2007. [Consultado 03 Marzo de 2007]. Disponible Internet: http://www.infoplcn.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.pdf

Las paredes del tanque absorben parte de la radiación y al detector llega sólo un pequeño porcentaje. Los detectores son, en general, detectores de cámara iónica

y utilizan amplificadores de c.c. o de c.a. El instrumento dispone de compensación de temperatura, de linealización de la señal de salida, y de reajuste de la pérdida de actividad de la fuente de radiación. Como desventajas en su aplicación figuran el blindaje de la fuente y el cumplimiento de las leyes sobre protección de radiación.

La precisión en la medida es de $\pm 0,5$ a ± 2 %, y el instrumento puede emplearse para todo tipo de líquidos ya que no está en contacto con el proceso. Su lectura viene influida por el aire o los gases disueltos en el líquido. El sistema se emplea en caso de medida de nivel en tanques de acceso difícil o peligroso. Es ventajoso cuando existen presiones elevadas en el interior del tanque que impiden el empleo de otros sistemas de medición. Hay que señalar que el sistema es caro y que la instalación no debe ofrecer peligro alguno de contaminación radiactiva siendo necesario señalar debidamente las áreas donde están instalados los instrumentos y realizar inspecciones periódicas de seguridad.

► **Medidor de Nivel Láser.** Se utiliza en aplicaciones donde las condiciones son muy duras, y donde los instrumentos de nivel convencionales fallan; tal es el caso de la medición de metal fundido, donde la medida del nivel debe realizarse sin contacto con el líquido y a la mayor distancia posible por existir unas condiciones de calor extremas. El sistema consiste en un rayo láser enviado a través de un tubo de acero y dirigido por reflexión en un espejo sobre la superficie del metal fundido. El aparato mide el tiempo que transcurre entre el impulso emitido y el impulso de retorno que es registrado en un foto detector de alta resolución, y este tiempo es directamente proporcional a la distancia del aparato emisor a la superficie del metal en fusión, es decir, da la lectura del nivel.

1.2.3. Digitalización. La digitalización o conversión analógica-digital (conversión A/D) consiste básicamente en realizar de forma periódica medidas de la amplitud de la señal y traducirlas a un lenguaje numérico. La conversión A/D también es conocida por el acrónimo inglés ADC (analogic to digital conversion). En esta definición están los cuatro procesos que intervienen en la conversión analógica-digital.

Muestreo: El muestreo (en inglés, sampling) consiste en tomar muestras periódicas de la amplitud de onda. La velocidad con que se toman esta muestra, es decir, el número de muestras por segundo, es lo que se conoce como frecuencia de muestreo.

Retención: En inglés, Hold, Las muestras tomadas han de ser retenidas (retención) por un circuito de retención (Hold), el tiempo suficiente para permitir evaluar su nivel (cuantificación). Desde el punto de vista matemático este proceso no se contempla ya que se trata de un recurso técnico debido a limitaciones prácticas y carece, por tanto, de modelo matemático.

Cuantificación: En el proceso de cuantificación se mide el nivel de voltaje de cada una de las muestras. Consiste en asignar un margen de valor de una señal analizada a un único nivel de salida. Incluso en su versión ideal, añade, como resultado, una señal indeseada a la señal de entrada: el ruido de cuantificación.

Codificación: La codificación consiste en traducir los valores obtenidos durante la cuantificación al código binario. Hay que tener presente que el código binario es el más utilizado, pero también existen otros tipos de códigos que también son utilizados. Durante el muestreo y la retención, la señal aun es analógica puesto que aún puede tomar cualquier valor. No obstante, a partir de la cuantificación, cuando la señal ya toma valores finitos, la señal ya es digital. Los cuatro procesos tienen lugar en un conversor analógico-digital.

► **Conversor Análogo/Digital.** En el control automático y monitoreo resulta conveniente efectuar las funciones de regulación de sistemas mediante técnicas digitales, sin embargo en la gran mayoría de los procesos la señal disponible es analógica, ya que son muchos los transductores que poseen una salida eléctrica de tipo analógica, correspondiente a la magnitud de la variable de medición como pueden ser las señales de audio, video, los puentes de medición, las celdas extensiométricas, los termopares, etc.

Estas características obligan a tener que efectuar una conversión analógica digital (ADC), por otro lado puede ser necesario actuar analógicamente sobre un controlador o elemento de control final, o efectuar una representación analógica sobre un registrador, una pantalla, etc. lo que nos obliga a realizar la conversión inversa, esto es, digital analógica (DAC), es necesario pues disponer de estos elementos capaces de efectuar esta conversión en uno u otro sentido, con unas características de velocidad adecuadas a cada caso. La conversión analógica digital es el proceso por el cual una magnitud analógica se convierte al formato digital. La conversión A/D es necesaria cuando se debe expresar en forma digital una serie de magnitudes medidas, para procesarlas en una computadora, presentarlas en un display o almacenarlas.

○ **Teoría básica de los adc.** Un convertidor analógico-digital toma un voltaje de entrada analógico y después de cierto tiempo produce un código con salida digital que representa a la entrada analógica con una precisión y resolución determinada.

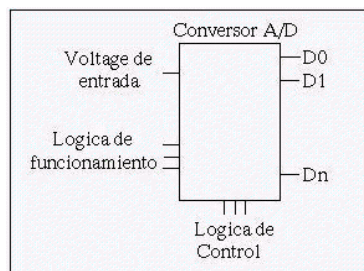
- Características Básicas.
 - Impedancia de entrada
 - Rango de entrada
 - Número de bits
 - Resolución
 - Tensión de fondo de escala
 - Tiempo de conversión
 - Error de conversión

Tiempo de conversión: Es el tiempo que tarda en realizar una medida el convertidor en concreto, y dependerá de la tecnología de medida empleada. Evidentemente nos da una cota máxima de la frecuencia de la señal a medir. Este tiempo se mide como el transcurrido desde que el convertidor recibe una señal de inicio de conversión (normalmente llamada SOC, Start of Conversión) hasta que en la salida aparece un dato válido. Para tener constancia de un dato válido hay dos caminos:

- Esperar el tiempo de conversión máximo que aparece en la hoja de características.
- Esperar a que el convertidor nos envíe una señal de fin de conversión.

Si no se espera el tiempo de conversión correspondiente, en la salida se registrara un valor, que dependiendo de la constitución del convertidor será: Un valor aleatorio, como consecuencia de la conversión en curso el resultado de la última conversión (Ver Figura 10).

Figura 10. Esquema general ADC.



Fuente: VIERA CHILE, Iremis. Sistemas de Adquisición de Datos [en línea]. Cuba: Universidad "Hermanos Saiz Montes de Oca". Modificada por última vez 29 diciembre 2004. [Consultado 05 Noviembre de 2007]. Disponible en Internet: <http://www.monografias.com/trabajos17/sistemas-adquisicion-dato/sistemas-adquisicion-dato.shtml>

El proceso de conversión AD es generalmente mas completo y largo que el proceso inverso DA, se han creado y utilizado muchos métodos de conversión AD como son:

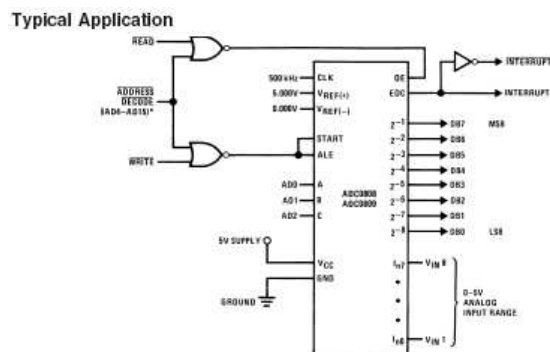
- Escalera
- Rampa Digital Simple
- Doble Rampa
- Rampa Digital Bipolar
- Ráfaga
- Conversión Serie Paralelo
- Aproximaciones sucesivas

- Método de Aproximaciones Sucesivas. Uno de los metodos mas usados es el metodo de aproximaciones sucesivas el cual consiste en determinar el valor de la cadena o palabra de unos y ceros que representa el valor de voltaje de entrada en numeros binarios.

El proceso de comparación empieza determinando el valor del bit mas significativo (MSB), esta comparación se efectua entre la tension de entrada V_x y la tension $V_h = V_{\max}/2$, correspondiente al código binario "1000", si V_h es mayor o igual a V_x , se determina que el bit de mayor peso debe ser uno, pero si $V_h < V_x$, se debe memorizar un cero en el registro de salida.

El registro de aproximaciones sucesivas esta constituido básicamente por un contador y un decodificador, que efectúa el direccionamiento de los "n" bi estables del registro de salida.

Figura 11. Diagrama del ADC0808.



Los convertidores de aproximaciones sucesivas tienen la precisión del conversor DAC que contienen y poseen un tiempo de conversión que puede llegar a ser del orden de 0.1 μ s, estos bajos tiempos de conversión y su relativamente bajo costo, hacen que los convertidores de 8, 10 y 12 bits sean los más utilizados.

Conociendo el número de Bits, se puede encontrar el máximo número de palabras diferentes que puede proporcionar la salida digital.

La Resolución es entonces $= 2^n$

Donde n = número de bits. Así un convertidor de 8 bits sólo podrá dar a la salida $2^8 = 256$ valores posibles.

Existe otra resolución que se define como la razón de cambio del valor en el voltaje de entrada, $V_{entrada}$. Que se requiere para cambiar en 1LSB la salida digital. Esto es cuando se conoce el valor de V_{in} , a escala completa. El voltaje de entrada a escala completa V_{in} , es proporcionado por el fabricante en sus hojas de especificaciones.

Entonces V_{in} es el valor máximo presente en la entrada analógica, para proporcionar unos lógicos en todas las salidas de Bit digitales.

$$Resolucion = \frac{V_{in}}{2^n - 1}$$

Restando solo la manera de encontrar una ecuación de entrada – salida, para facilitar rápidamente la palabra digital, incluso en forma decimal, que entrega el convertidor.

$$D = \frac{V_{entrada}}{Resolucion}$$

$V_{entrada}$ = voltaje analógico presente en el instante
 D = Valor decimal de la salida digital

Por ejemplo, un convertidor de 8 bits con una tensión de fondo de escala de 2V tendrá una resolución de:

$$\frac{2V}{2^8 - 1} = 0.0784 \frac{mV}{\text{paso}}$$

En cambio, para el mismo convertidor, si se cambia la tensión de referencia, y por tanto la de fondo de escala, la resolución será de:

$$\frac{5V}{2^8 - 1} = 0.019 \frac{mV}{\text{Paso}}$$

1.2.4. Microcontrolador. En 1980 aproximadamente, los fabricantes de circuitos integrados iniciaron la difusión de un nuevo circuito para control, medición e instrumentación al que llamaron microcomputador en un sólo chip o de manera más exacta microcontrolador. Un microcontrolador es un circuito integrado que contiene toda la estructura (arquitectura) de un microcomputador, o sea CPU, RAM, ROM y circuitos de entrada y salida. Los resultados de tipo práctico, que pueden lograrse a partir de estos elementos, son sorprendentes.

► Ventajas de un microcontrolador / microprocesador. Las diferencias más importantes entre un microcontrolador / microprocesador son:

Por ejemplo la configuración mínima básica de un microprocesador estaba constituida por un Micro de 40 Pines, Una memoria RAM de 28 Pines, una memoria ROM de 28 Pines y un decodificador de direcciones de 18 pines; una de las principales características que tiene un microcontrolador es que incluye todo estos elementos en un solo Circuito Integrado implicando una gran ventaja en varios factores:

- En el circuito impreso por su amplia simplificación de circuitería,
- el costo para un sistema basado en microcontrolador es mucho menor
- el tiempo de desarrollo de un proyecto se disminuye considerablemente.

Los microcontroladores se basan principalmente en dos arquitecturas: Las arquitecturas RISC y CISC son ejemplos de CPU con un conjunto de instrucciones para arquitecturas basadas en registros.

► Arquitecturas.

- Arquitecturas CISC. La microprogramación es una característica importante y esencial de casi todas las arquitecturas CISC. La microprogramación significa que cada instrucción de máquina es interpretada por un microprograma localizado en la memoria del circuito integrado del procesador.

En la década de los sesentas la microprogramación, por sus características, era la técnica más apropiada para las tecnologías de memorias existentes en esa época y permitía desarrollar también procesadores con compatibilidad ascendente. En consecuencia, los procesadores se dotaron de poderosos conjuntos de instrucciones. Las instrucciones compuestas son decodificadas internamente y ejecutadas con una serie de microinstrucciones almacenadas en una ROM interna. Para esto se requieren de varios ciclos de reloj (al menos uno por microinstrucción).

- Arquitecturas RISC. Buscando aumentar la velocidad del procesamiento se descubrió en base a experimentos que, con una determinada arquitectura de base, la ejecución de programas compilados directamente con microinstrucciones y residentes en memoria externa al circuito integrado resultaban ser mas eficientes, gracias a que el tiempo de acceso de las memorias se fue decrementando conforme se mejoraba su tecnología de encapsulado.

Debido a que se tiene un conjunto de instrucciones simplificado, éstas se pueden implantar por hardware directamente en la CPU, lo cual elimina el microcódigo y la necesidad de decodificar instrucciones complejas.

Las arquitecturas von neumann y harvard son arquitecturas de CPU caracterizadas por la forma en que se almacenan y procesan los datos.

- Arquitectura Von Neumann. Se refiere a las arquitecturas de computadoras que utilizan el mismo dispositivo de almacenamiento tanto para las instrucciones como para los datos. Los ordenadores con arquitectura Von Neumann constan de cinco partes:

- La unidad aritmético-lógica o ALU
- La unidad de control
- La memoria
- Un dispositivo de entrada/salida.
- El bus de datos que proporciona un medio de transporte de los datos entre las distintas partes.

Un ordenador con arquitectura Von Neumann realiza o emula los siguientes pasos secuencialmente:

- Obtiene la siguiente instrucción desde la memoria en la dirección indicada por el contador de programa y la guarda en el registro de instrucción.
- Aumenta el contador de programa en la longitud de la instrucción para apuntar a la siguiente.
- Decodifica la instrucción mediante la unidad de control. Ésta se encarga de coordinar el resto de componentes del ordenador para realizar una función determinada.

Se ejecuta la instrucción. Ésta puede cambiar el valor del contador del programa, permitiendo así operaciones repetitivas. El contador puede cambiar también cuando se cumpla una cierta condición aritmética, haciendo que el ordenador pueda 'tomar decisiones', que pueden alcanzar cualquier grado de complejidad, mediante la aritmética y lógica anteriores. Vuelve al paso 1, la arquitectura tradicional de computadoras y microprocesadores está basada en la arquitectura Von Neumann, en la cual la unidad central de proceso (CPU), está conectada a una memoria única donde se guardan las instrucciones del programa y los datos.

- Arquitectura harvard. Tiene la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes. Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), la otra sólo almacena datos (Memoria de Datos).

Ambos buses son totalmente independientes y pueden ser de distintos anchos. Para un procesador de Set de Instrucciones Reducido, o RISC (Reduced Instrucción Set Computer), el set de instrucciones y el bus de memoria de programa pueden diseñarse de tal manera que todas las instrucciones tengan una sola posición de memoria de programa de longitud. Además, al ser los buses independientes, la CPU puede acceder a los datos para completar la ejecución de una instrucción, y al mismo tiempo leer la siguiente instrucción a ejecutar.

- Ventajas de esta arquitectura:
 - El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.

- El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad en cada operación.

Una de las desventajas de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontraran físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador).

Algunos microcontroladores más especializados poseen además conversores análogo digital, temporizadores, contadores y un sistema para permitir la comunicación en serie y en paralelo. Se pueden crear muchas aplicaciones con los microcontroladores. Estas aplicaciones de los microcontroladores son ilimitadas (el límite es la imaginación) entre ellas podemos mencionar: sistemas de alarmas, juego de luces, paneles publicitarios, etc. Controles automáticos para la Industria en general. Entre ellos control de motores DC/AC y motores de paso a paso, control de máquinas, control de temperatura, control de tiempo, adquisición de datos mediante sensores, etc. Los Microcontroladores son dispositivos que se especializa en aplicaciones industriales y en ambientes electrónicos adversos.

El mismo es una integración de subsistemas que anteriormente formaban unidades especializadas e independientes, pero unidas por una ruta de un circuito impreso con el microprocesador forman lo que se conoce como un sistema, por lo que el microcontrolador va a integrar los siguientes subsistemas, aunque no siempre presenta todos los relacionados:

- CPU(Unidad de Procesamiento Central)
- Memoria RAM
- Memoria ROM o EPROM
- Entradas / Salidas
- Contadores / Temporizadores
- Conversores análogo / digital y digital / analógico
- Gestión de Interrupciones.

También se puede decir que el microcontrolador tiene una serie de características que son muy fundamentales a la hora de hacer una aplicación como son:

- Capacidad de Proceso de n bit.
- Circuito de Reloj Incorporado.
- Altas Frecuencias de Reloj según el micro.
- Múltiples Puertos de Entrada- Salida.

- Bajo Consumo (Power down) en las versiones CHMOS.
- Alta inmunidad al ruido eléctrico.
- Ampliación del set de instrucciones con algunas muy potentes para la adquisición y tratamiento de datos, tablas, multiplicación, división, etc.
- Instrucciones lógicas y de bifurcación orientadas al proceso de señales bit a bit (procesador booleano).
- Espacios de separados (memoria segregada).
- Protección de la memoria de programa (encriptación).
- Perro guardián (watchdog), que vigila el funcionamiento óptimo de la CPU.
- Posibilidades de comunicación estándar con otros sistemas (tipo full duplex).
- Conversores A/D integrados en el propio dispositivo.
- Salida de modulación de ancho del pulso (PWM) para conversión D/A.
- Multiplexores integrados en el dispositivo.
- Dos o tres temporizadores/ contadores de n bit
- Varias fuentes de interrupción programables con niveles de prioridad.

► Fabricantes. Fabricantes de μC más conocidas en el mercado actualmente son (Ver Tabla 2).

Tabla 2. Fabricantes de μC más conocidas en el mercado.

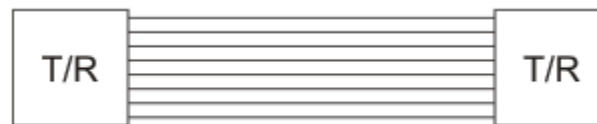
Motorola	Intel
Microchip	Zilog
National Semiconductor	Mitsubishi Electric
Rockwell	Dallas Semiconductor
SGS-Thomson	Texas Instruments
AMD	Atmel
OKI	Temec
Siemens	Philips
Fujitsu	NEC

1.2.5. Transmisión de datos. La comunicación en los medios informáticos se realiza de dos maneras:

► Transmisión en Paralelo. Todos los bits se transmiten simultáneamente, existiendo luego un tiempo antes de la transmisión del siguiente bloque. Este tipo de transmisión tiene lugar en el interior de una máquina o entre máquinas cuando la distancia es muy corta. La principal ventaja de este modo de transmitir datos es

la velocidad de transmisión y la mayor desventaja es el costo. También puede llegar a considerarse una transmisión en paralelo, aunque se realice sobre una sola línea, al caso de multiplexación de datos, donde los diferentes datos se encuentran intercalados durante la transmisión (Ver Figura 12).

Figura 12. Transmisión en paralelo.



► **Transmisión en Serie.** A la salida de una maquina los datos en paralelo se convierten los datos en serie, los mismos se transmiten y luego en el receptor tiene lugar el proceso inverso, volviéndose a obtener los datos en paralelo. La secuencia de bits transmitidos es por orden de peso creciente y generalmente el último bit es de paridad. Un aspecto fundamental de la transmisión serie es el sincronismo, entendiéndose como tal al procedimiento mediante el cual transmisor y receptor reconocen los ceros y unos de los bits de igual forma. El sincronismo puede tenerse a nivel de bit, de byte o de bloque, donde en cada caso se identifica el inicio y finalización de los mismos. La transmisión serie se presenta de dos formas sincronía y asincrónica.

► **Transmisión asincrónica.** Es también conocida como Star/stop. Requiere de una señal que identifique el inicio del carácter y a la misma se la denomina bit de arranque. También se requiere de otra señal denominada señal de parada que indica la finalización del carácter o bloque. En este caso los n bits que componen un mensaje se transmiten uno detrás de otro por la misma línea.

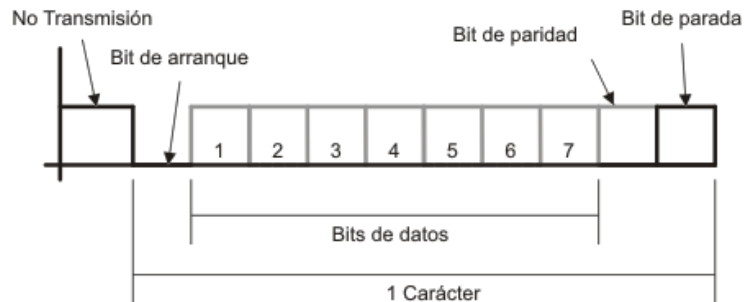
Figura 13. Transmisión en serie.



Generalmente cuando no hay transmisión, una línea se encuentra en un nivel alto. Tanto el transmisor como el receptor, saben cual es la cantidad de bits que componen el carácter son 7 (Ver Figura 13). Los bits de parada son una manera de fijar qué delimita la cantidad de bits del carácter y cuando e transmite un

conjunto de caracteres, luego de los bits de parada existe un bit de arranque entre los distintos caracteres. A pesar de ser una forma comúnmente utilizada, la desventaja de la transmisión asincrónica es su bajo rendimiento, puesto que como en el caso del ejemplo, el carácter tiene 7 bits pero para efectuar la transmisión se requieren 10. O sea que del total de bits transmitidos solo el 70% pertenecen a datos (Ver Figura 14).

Figura 14. Formato de un carácter.



► **Transmisión Sincrónica.** En este tipo de transmisión es necesario que el transmisor y el receptor utilicen la misma frecuencia de reloj en ese caso la transmisión se efectúa en bloques, debiéndose definir dos grupos de bits denominados delimitadores, mediante los cuales se indica el inicio y el fin de cada bloque. Este método es más efectivo por que el flujo de información ocurre en forma uniforme, con lo cual es posible lograr velocidades de transmisión más altas. Para lograr el sincronismo, el transmisor envía una señal de inicio de transmisión mediante la cual se activa el clock del receptor. A partir de dicho instante transmisor y receptor se encuentran sincronizados. Otra forma de lograr el sincronismo es mediante la utilización de códigos auto sincronizantes los cuales permiten identificar el inicio y el fin de cada bit.

► **Comunicación Serial.** El estándar de comunicación serial mas conocido y usado es el **RS-232** es una interfaz que designa una norma para el intercambio serie de datos binarios entre un DTE y un DCE.

Todas las normas RS-232 cumplen con los siguientes niveles de Voltaje:

- Un "1" lógico es un voltaje comprendido entre $-5v$ y $-15v$ en el transmisor y entre $-3v$ y $-25v$ en el receptor.
 - Un "0" lógico es un voltaje comprendido entre $+5v$ y $+15v$ en el transmisor y entre $+3v$ y $+25v$ en el receptor.
- La interfaz RS-232 está diseñada para distancias cortas, de unos 15 metros o menos, y para velocidades de comunicación bajas, de no más de 20 Kb. A pesar

de ello, muchas veces se utiliza a mayores velocidades con un resultado aceptable. La interfaz puede trabajar en comunicación asíncrona o síncrona y tipos de canal simplex, half duplex o full duplex. En un canal simplex los datos siempre viajarán en una dirección, por ejemplo desde DCE a DTE. En un canal half duplex, los datos pueden viajar en una u otra dirección, pero sólo durante un determinado periodo de tiempo; luego la línea debe ser conmutada antes que los datos puedan viajar en la otra dirección. En un canal full duplex, los datos pueden viajar en ambos sentidos simultáneamente. Las líneas de handshaking de la RS-232 se usan para resolver los problemas asociados con este modo de operación, tal como en qué dirección los datos deben viajar en un instante determinado.

Para la sincronización de una comunicación se precisa siempre de una línea adicional a través de la cual el emisor y el receptor intercambian la señal del pulso. Pero en la transmisión serie a través de un cable de dos líneas esto no es posible ya que ambas están ocupadas por los datos y la masa. Por este motivo se intercalan antes y después de los datos informaciones de estado según el protocolo RS-232. Esta información es determinada por el emisor y receptor al estructurar la conexión mediante la correspondiente programación de sus puertos serie. Esta información puede ser la siguiente:

- Bit de paridad: con este bit se pueden descubrir errores en la transmisión. Se puede dar paridad par o impar. En la paridad par, por ejemplo, la palabra de datos a transmitir se completa con el bit de paridad de manera que el número de bits 1 enviados se par.
- Bit de parada: indica la finalización de la transmisión de una palabra de datos. El protocolo de transmisión de datos permite 1, 1.5 y 2 bits de parada.
- Bit de inicio: cuando el receptor detecta el bit de inicio sabe que la transmisión ha comenzado y es a partir de entonces que debe leer la transmisión ha comenzado y es a partir de entonces que debe leer las señales de la línea a distancias concretas de tiempo, en función de la velocidad determinada.

Cuando un dispositivo de los que están conectados a una interfaz RS-232 procesa los datos a una velocidad menor de la que los recibe deben de conectarse las líneas handshaking que permiten realizar un control de flujo tal que al dispositivo más lento le de tiempo de procesar la información. Las líneas de "hand shaking" que permiten hacer este control de flujo son las líneas RTS y CTS. Los diseñadores del estándar no concibieron estas líneas para que funcionen de este modo, pero dada su utilidad en cada interfaz posterior se incluye este modo de uso. La interfaz RS-232 dispone de hasta 25 líneas que están orientadas a la comunicación de dos equipos PC (DTE) a través de módems (DCE). En este caso se utilizarán para la conexión de los equipos PC prescindiendo de los módems. Para ello de las 25 líneas que posee se han utilizado sólo las siguientes:

- Línea de transmisión de datos (TxD): línea por la que el DTE (PC) envía los datos.
- Línea de recepción de datos (RxD): línea por la que el DTE (PC) recibe los datos.
- DTE preparado (DTR): línea por la que el DTE (PC) indica al DCE (módem) que está activo para comunicarse con el módem.
- DCE preparado (DSR): línea por la que el DCE (módem) indica al DTE (PC) que está activo para establecer la comunicación.
- Petición de envío (RTS): con esta línea, el DTE (PC) indica al DCE (módem) que está preparado para transmitir datos.
- Preparado para enviar (CTS): tras un RTS, el DCE (módem) pone esta línea en 1 lógico, tan pronto como está preparado para recibir datos.
- Masa: necesaria para que tenga lugar la transmisión.

Estas líneas son controladas mediante la programación de los registros de la UART que es un chip especial para la entrada y salida de caracteres y, sobre todo, para la conversión de palabras de datos en las correspondientes señales del puerto serie.

► Limitaciones de la RS-232. Cuando una señal cambia de una condición a otra, la especificación limita el tiempo que puede permanecer en la región de transición. Este requerimiento determina el máximo de capacidad (eléctrica) distribuida admisible en el cable, porque la capacidad limita el tiempo de transición de la señal. La norma RS-232 especifica que la capacidad en la línea no debe superar los 2.500 picofaradios. Los cables que se suelen utilizar tienen una capacidad de 120 a 150 picofaradios por metro de longitud, por lo que la RS-232 tiene como límite de 15 m de distancia. Una segunda limitación de la RS-232 es su método de toma de tierra o retorno común. Este método, llamado transmisión no balanceada, funciona bien la mayor parte del tiempo. Sin embargo, si hay diferencia de potencial entre los dos extremos del cable (lo cual es bastante probable en recorridos largos), se reduce la región de transición entre marca y espacio. Cuando ocurre esto, existe la posibilidad que no se interpreten bien los distintos estados de la señal.

Otra dificultad es su máximo de 20 KB/s para la velocidad de transmisión. Si bien en el momento de aparición del estándar era suficiente, en la actualidad, comparando con las velocidades alcanzadas por las redes de área local, 10 y 100 MB/s y las exigencias de ancho de banda que las aplicaciones requieren, la RS-232 C en algunos casos está disminuyendo su aplicación.

A partir de la RS-232 se desarrollaron nuevas interfaces que pretenden transmitir a mayor velocidad alcanzando mayor distancia. Estas nuevas interfaces como la RS-422 y la RS-423 eliminan algunas de las restricciones de la RS-232, por

ejemplo, la de poseer un retorno común para todas las señales.

► **Puerto Serie.** Un puerto serie es una interfaz de comunicaciones entre ordenadores y periféricos en donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits a la vez. Entre el puerto serie y el puerto paralelo, existe la misma diferencia que entre una carretera tradicional de un sólo carril por sentido y una autovía con varios carriles por sentido. En informática, un puerto serie es una interfaz física de comunicación en serie a través de la cual se transfiere información mandando o recibiendo un bit cada vez, en contraste con un puerto paralelo.

A lo largo de la mayor parte de la historia de las computadoras, la transferencia de datos a través de los puertos de serie conectaba las computadoras a dispositivos como terminales o módems. Los mouse, teclados, y otros periféricos también se conectaban de esta forma. Mientras que otras interfaces como Ethernet, FireWire, y USB mandaban datos como un flujo en serie, el término "puerto de serie" normalmente identifica el hardware más o menos conforme al estándar RS-232, diseñado para interactuar con un módem o con un dispositivo de comunicación similar. El puerto serie RS-232 (también conocido como COM) por excelencia es el que utiliza cableado simple desde 3 hilos hasta 25 y que conecta ordenadores o microcontroladores a todo tipo de periféricos, desde terminales a impresoras y módems pasando por ratones.

► **Comunicación USB.** El Universal Serial Bus es una interfaz que provee un estándar de bus serie para conectar dispositivos a un computador. Un sistema USB tiene un diseño asimétrico, que consiste en un solo servidor y múltiples dispositivos conectados en serie para ampliar la gama de conexión, en una estructura de árbol utilizando concentradores especiales. Se pueden conectar hasta 127 dispositivos a un sólo servidor, pero la suma debe incluir a los concentradores también, así que el total de dispositivos realmente usables es algo menor. Esta interfaz utiliza cuatro líneas de las cuales dos son para alimentación (+5V y GND), las dos restantes son utilizadas para el transporte de señales diferenciales usando una codificación NRZI con un campo de sincronización para los relojes del Host y el dispositivo. Esta especificación soporta Plug n Play con carga y descarga dinámica de drivers, lo cual permite que el usuario simplemente conecte un dispositivo al bus de datos, el Host automáticamente detectara un nuevo dispositivo, lo interrogara y cargara el controlador mas apropiado.

1.2.6. Familia de protocolos de internet. La familia de protocolos de Internet es un conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. En ocasiones se la denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de

Internet (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia. Existen tantos protocolos en este conjunto que llegan a ser más de 100 diferentes, entre ellos se encuentra el popular HTTP (HyperText Transfer Protocol), que es el que se utiliza para acceder a las páginas web, además de otros como el ARP (Address Resolution Protocol) para la resolución de direcciones, el FTP (File Transfer Protocol) para transferencia de archivos, y el SMTP (Simple Mail Transfer Protocol) y el POP (Post Office Protocol) para correo electrónico, TELNET para acceder a equipos remotos, entre otros. El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN). TCP/IP fue desarrollado y demostrado por primera vez en 1972 por el departamento de defensa de los Estados Unidos, ejecutándolo en ARPANET, una red de área extensa del departamento de defensa. Una interpretación simplificada de la pila TCP/IP se muestra en la (Ver Tabla 3).

Tabla 3. Modelo TCP/IP.

5	Aplicación	ej. HTTP, FTP, DNS (protocolos de enrutamiento como BGP y RIP, que por varias razones funcionen sobre TCP y UDP respectivamente, son considerados parte del nivel de red)
4	Transporte	ej. TCP, UDP, RTP, SCTP (protocolos de enrutamiento como OSPF, que funcionen sobre IP, son considerados parte del nivel de red)
3	Internet	Para TCP/IP este es el Protocolo de Internet (IP) (protocolos requeridos como ICMP e IGMP funcionan sobre IP, pero todavía se pueden considerar parte del nivel de red; ARP no funciona sobre IP)
2	Enlace	ej. Ethernet, Token Ring, PPP, HDLC, Frame Relay, RDSI, ATM, IEEE 802.11, FDDI
1	Físico	ej. medio físico, y técnicas de codificación, T1, E1

Fuente: Familia de protocolos de Internet [en línea]. Florida: Wikimedia Foundation, 2006. Modificada por última vez 7 febrero 2008.. [Consultado 11 Noviembre de 2007]. Disponible en Internet:

http://es.wikipedia.org/wiki/Familia_de_protocolos_de_Internet

► Protocolo de Internet. El Protocolo de Internet (IP, de sus siglas en inglés *Internet Protocol*) es un protocolo orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados. Los datos en una red basada en IP son enviados en bloques conocidos como paquetes o data gramas (en el protocolo IP estos términos se suelen usar indistintamente). En particular, en IP no se necesita ninguna configuración antes de que un equipo intente enviar paquetes a otro con el que no se había comunicado antes. El Protocolo de Internet provee un servicio

de datos gramas no fiable (también llamado del *mejor esfuerzo (best effort)*, lo hará lo mejor posible pero garantizando poco). IP no provee ningún mecanismo para determinar si un paquete alcanza o no su destino y únicamente proporciona seguridad (mediante *checksums* o sumas de comprobación) de sus cabeceras y no de los datos transmitidos. Las cabeceras IP contienen las direcciones de las máquinas de origen y destino (direcciones IP), direcciones que serán usadas por los conmutadores de paquetes (switches) y los enrutadores (routers) para decidir el tramo de red por el que reenviarán los paquetes.

► **Protocolo TCP.** Es uno de los protocolos fundamentales en Internet. Fue creado entre los años 1973 - 1974 por Vint Cerf y Robert Kahn). Muchos programas dentro de una red de datos compuesta por ordenadores pueden usar TCP para crear *conexiones* entre ellos a través de las cuales enviarse un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto. TCP da soporte a muchas de las aplicaciones más populares de Internet, incluidas HTTP, SMTP y SSH.

○ **Formato de los segmentos TCP.** En el nivel de transporte, los paquetes de bits que constituyen las unidades de datos de protocolo o PDU (protocol data unit) se llaman segmentos cada uno tiene un número determinado de bits como se puede observar en la (Ver Tabla 4). Formato de los segmentos TCP.

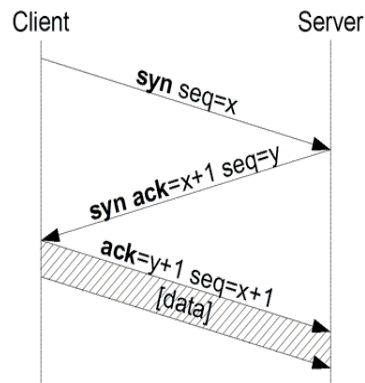
Tabla 4. Formato de los segmentos TCP.

Nbits	Bits 0 - 3	4 - 7	8 - 15	16 - 31
0	Puerto Origen			Puerto Destino
32	Número de Secuencia			
64	Número de Acuse de Recibo (ACK)			
96	Longitud cabecera TCP	Reservado	Flags	Ventana
128	Suma de Verificación (Checksum)			Puntero Urgente
160	Opciones + Relleno (opcional)			
224	Datos			

Fuente: Transmission Control Protocol [en línea]. Florida: Wikimedia Foundation, 2008. Modificada por última vez 30 enero 2008. [Consultado 11 Noviembre de 2007]. Disponible en Internet: <http://es.wikipedia.org/wiki/TCP>

- Establecimiento de la conexión. Las aplicaciones envían flujos de bytes a la capa TCP para ser enviados a la red. TCP divide el flujo de bytes llegado de la aplicación en segmentos de tamaño apropiado (normalmente esta limitación viene impuesta por la MTU del nivel de enlace de datos de la red a la que la entidad está asociada) y le añade sus cabeceras. TCP pasa el segmento resultante a la capa IP, donde a través de la red, llega a la capa TCP de la entidad destino.
- TCP comprueba que ningún segmento se ha perdido dando a cada uno un *número de secuencia*, que es también usado para asegurarse de que los paquetes han llegado a la entidad destino en el orden correcto. TCP devuelve un asentimiento por bytes que han sido recibidos correctamente; un temporizador en la entidad origen del envío causará un *timeout* si el asentimiento no es recibido en un tiempo razonable, y el (presuntamente desaparecido) paquete será entonces *retransmitido*. TCP revisa que no haya bytes dañados durante el envío usando un checksum; es calculado por el emisor en cada paquete antes de ser enviado, y comprobado por el receptor. El campo Flags utiliza 8 bits para activar o desactivar cada una de las siguientes funciones: URG, ACK, PSH, Flag RST, SYN y FIN.

Figura 15. Establecimiento de la conexión.

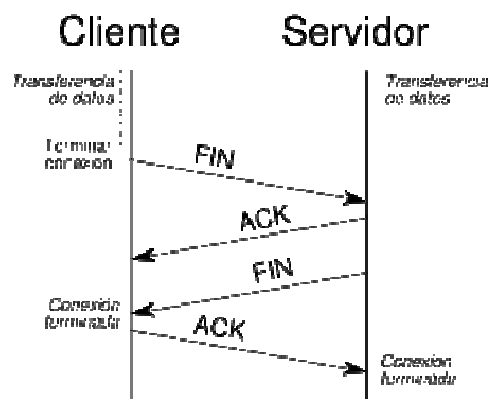


Fuente: Familia de protocolos de Internet [en línea]. Florida: Wikimedia Foundation, 2008. Modificada por última vez 30 enero 2008. [Consultado 11 Noviembre de 2007]. Disponible en Internet: <http://es.wikipedia.org/wiki/TCP>

El lado cliente de una conexión realiza una apertura activa de un puerto enviando un segmento SYN inicial al servidor como parte de la negociación en tres pasos. El lado servidor respondería a la petición SYN válida con un paquete SYN/ACK. Finalmente, el cliente debería responderle al servidor con un ACK, completando así la negociación en tres pasos (SYN, SYN/ACK y ACK) y la fase de establecimiento de conexión (Ver Figura 15).

Fin de la conexión. La fase de finalización de la conexión usa una negociación en cuatro pasos (four-way handshake), terminando la conexión desde cada lado independientemente. Cuando uno de los dos extremos de la conexión desea parar su "mitad" de conexión transmite un paquete FIN, que el otro interlocutor asentirá con un ACK. Por tanto, una desconexión típica requiere un par de segmentos FIN y ACK desde cada lado de la conexión. Una conexión puede estar "medio abierta" en el caso de que uno de los lados la finalice pero el otro no. El lado que ha dado por finalizada la conexión no puede enviar más datos pero la otra parte si podrá (Ver Figura 16).

Figura 16. Cierre de una conexión según el estándar.



Fuente: Familia de protocolos de Internet [en línea]. Florida: Wikimedia Foundation, 2006. Modificada por última vez 30 enero 2008. [Consultado 11 Noviembre de 2007]. Disponible en Internet: <http://es.wikipedia.org/wiki/TCP>

- Puertos TCP. TCP usa el concepto de número de puerto para identificar a las aplicaciones emisoras y receptoras. Cada lado de la conexión TCP tiene asociado un número de puerto (de 16 bits sin signo, con lo que existen 65536 puertos posibles) asignado por la aplicación emisora o receptora. Los puertos son clasificados en tres categorías: bien conocidos, registrados y dinámicos/privados. Los puertos bien conocidos son asignados por la Internet Assigned Numbers Authority (IANA), van del 0 al 1023 y son usados normalmente por el sistema o por procesos con privilegios. Las aplicaciones que usan este tipo de puertos son ejecutadas como servidores y se quedan a la escucha de conexiones. Algunos ejemplos son: FTP (21), SSH (22), Telnet (23), SMTP (25) y HTTP (80).
- Modelo TCP/IP. El TCP / IP es la base del Internet que sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local y área extensa.

- Nivel físico. El nivel físico describe las características físicas de la comunicación, como las convenciones sobre la naturaleza del medio usado para la y todo lo relativo a los detalles como los conectores, código de canales y modulación, potencias de señal, longitudes de onda, sincronización y temporización y distancias máximas.

- Nivel de enlace de datos. El nivel de enlace de datos especifica como son transportados los paquetes sobre el nivel físico, incluido los delimitadores (patrones de bits concretos que marcan el comienzo y el fin de cada trama). Ethernet, por ejemplo, incluye campos en la cabecera de la trama que especifican que máquina o máquinas de la red son las destinatarias de la trama. Ejemplos de protocolos de nivel de enlace de datos son Ethernet, Wireless Ethernet, SLIP, Token Ring y ATM.

- Nivel de internet. Como fue definido originalmente, el nivel de red soluciona el problema de conseguir transportar paquetes a través de una red sencilla. Ejemplos de protocolos son X.25 y Host/IMP Protocol de ARPANET. Con la llegada del concepto de Internet, nuevas funcionalidades fueron añadidas a este nivel, basadas en el intercambio de datos entre una red origen y una red destino. Generalmente esto incluye un enrutamiento de paquetes a través de una red de redes, conocida como Internet. En la familia de protocolos de Internet, IP realiza las tareas básicas para conseguir transportar datos desde un origen a un destino. IP puede pasar los datos a una serie de protocolos superiores; cada uno de esos protocolos es identificado con un único "Número de protocolo IP". ICMP y IGMP son los protocolos 1 y 2, respectivamente.

- Nivel de transporte. Los protocolos del nivel de transporte pueden solucionar problemas como la fiabilidad y la seguridad de que los datos llegan en el orden correcto. En el conjunto de protocolos TCP/IP, los protocolos de transporte también determinan a que aplicación van destinados los datos. Los protocolos de enrutamiento dinámico que técnicamente encajan en el conjunto de protocolos TCP/IP (ya que funcionan sobre IP) son generalmente considerados parte del nivel de red; un ejemplo es OSPF (protocolo IP número 89). TCP (protocolo IP número 6) es un mecanismo de transporte fiable y orientado a conexión, que proporciona un flujo fiable de bytes, que asegura que los datos llegan completos, sin daños y en orden. TCP realiza continuamente medidas sobre el estado de la red para evitar sobrecargarla con demasiado tráfico. Además, TCP trata de enviar todos los datos correctamente en la secuencia especificada. Esta es una de las principales diferencias con UDP, y puede convertirse en una desventaja en flujos en tiempo real (muy sensibles a la variación del retardo) o aplicaciones de enrutamiento con porcentajes altos de pérdida en el nivel de internet.

- UDP (protocolo IP número 17) es un protocolo de datagramas sin conexión. Es un protocolo no fiable (*best effort* al igual que IP) - no porque sea particularmente malo, sino porque no verifica que los paquetes lleguen a su destino, y no da garantías de que lleguen en orden. Si una aplicación requiere estas características, debe llevarlas a cabo por sí misma o usar TCP. UDP es usado normalmente para aplicaciones de streaming (audio, video, etc) donde la llegada a tiempo de los paquetes es más importante que la fiabilidad, o para aplicaciones simples de tipo petición/respuesta como el servicio DNS, donde la sobrecarga de las cabeceras que aportan la fiabilidad es desproporcionada para el tamaño de los paquetes. TCP y UDP son usados para dar servicio a una serie de aplicaciones de alto nivel. Las aplicaciones con una dirección de red dada son distinguibles entre sí por su número de puerto TCP o UDP. Por convención, los puertos bien conocidos (*well-known ports*) son asociados con aplicaciones específicas.

- Nivel de aplicación. El nivel de aplicación es el nivel que los programas más comunes utilizan para comunicarse a través de una red con otros programas. Los procesos que acontecen en este nivel son aplicaciones específicas que pasan los datos al nivel de aplicación en el formato que internamente use el programa y es codificado de acuerdo con un protocolo estándar. Algunos programas específicos se considera que se ejecutan en este nivel. Proporcionan servicios que directamente trabajan con las aplicaciones de usuario. Estos programas y sus correspondientes protocolos incluyen a HTTP (HyperText Transfer Protocol), FTP (Transferencia de archivos), SMTP (correo electrónico), SSH (login remoto seguro), DNS (Resolución de nombres de dominio) y a muchos otros. Una vez que los datos de la aplicación han sido codificados en un protocolo estándar del nivel de aplicación son pasados hacia abajo al siguiente nivel de la pila de protocolos TCP/IP.

- Características de TCP/IP. Los protocolos TCP/IP presentan las siguientes características:

- Son estándares de protocolos abiertos y gratuitos. Su desarrollo y modificaciones se realizan por consenso, no a voluntad de un determinado fabricante. Cualquiera puede desarrollar productos que cumplan sus especificaciones.

- Independencia a nivel software y hardware Su amplio uso los hace especialmente idóneos para interconectar equipos de diferentes fabricantes, no solo a Internet sino también formando redes locales. La independencia del hardware nos permite integrar en una sola varios tipos de redes (Ethernet, Token Ring, X.25...)

- Proporcionan un esquema común de direccionamiento que permite a un dispositivo con TCP/IP localizar a cualquier otro en cualquier punto de la red.

- Son protocolos estandarizados de alto nivel que soportan servicios al usuario y son ampliamente disponibles y consistentes.

- Ventajas e Inconvenientes con TCP/IP. El conjunto TCP/IP está diseñado para enrutar y tiene un grado muy elevado de fiabilidad, es adecuado para redes grandes y medianas, así como en redes empresariales. Se utiliza a nivel mundial para conectarse a Internet y a los servidores web. Es compatible con las herramientas estándar para analizar el funcionamiento de la red.

Uno de los inconvenientes que se presenta con TCP/IP es que es más difícil de configurar y de mantener que NetBEUI o IPX/SPX; además es algo más lento en redes con un volumen de tráfico medio bajo. Sin embargo, puede ser más rápido en redes con un volumen de tráfico grande donde haya que enrutar un gran número de tramas. El conjunto TCP/IP se utiliza tanto en redes empresariales como por ejemplo en campus universitarios o en complejos empresariales, en donde utilizan muchos enrutadores y conexiones a mainframe o a ordenadores UNIX, como así también en redes pequeñas o domésticas, y hasta en teléfonos móviles y en domótica.

- Direccionamiento IP y enrutamiento. Quizás los aspectos más complejos de IP son el direccionamiento y el enrutamiento. El direccionamiento se refiere a la forma como se asigna una dirección IP y como se dividen y se agrupan subredes de equipos. El enrutamiento consiste en encontrar un camino que conecte una red con otra y aunque es llevado a cabo por todos los equipos, es realizado principalmente por enrutadores que no son más que computadores especializados en recibir y enviar paquetes por diferentes interfaces de red, así como proporcionar opciones de seguridad, redundancia de caminos y eficiencia en la utilización de los recursos

- Dirección IP. Una dirección IP es un número que identifica de manera lógica y jerárquicamente a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo de Internet (Internet Protocol), que corresponde al nivel de red o nivel 3 del modelo de referencia OSI. Es habitual que un usuario que se conecta desde su hogar a Internet utilice una dirección IP. Esta dirección puede cambiar al reconectar; y a esta forma de asignación de dirección IP se denomina una dirección IP dinámica (normalmente se abrevia como IP dinámica). Los sitios de Internet que por su naturaleza necesitan estar permanentemente conectados, generalmente tienen una dirección IP fija (se aplica la misma reducción por IP fija o IP estática), es decir, no cambia con el tiempo. Los servidores de correo, dns, ftp públicos, servidores web necesariamente deben contar con una dirección IP fija o estática, ya que de esta forma se facilita su ubicación.

- Enrutamiento. En comunicaciones, el encaminamiento (a veces conocido por el anglicismo ruteo o enrutamiento) es el mecanismo por el que en una red los paquetes de información se hacen llegar desde su origen a su destino final, siguiendo un camino o ruta a través de la red. En una red grande o en un conjunto de redes interconectadas el camino a seguir hasta llegar al destino final puede suponer transitar por muchos nodos intermedios.

1.2.7. Lenguaje de programación. Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

► Lenguaje de máquina. Conjunto de instrucciones, comprendidas por el ordenador, escritas mediante bytes formados por ceros y unos que la máquina interpreta como un dato o una instrucción. Un simple patrón de bits, es hecho legible reemplazando valores crudos por símbolos denominados mnemónicos. Se inventó para facilitar la tarea de los primeros programadores que hasta ese momento tenían que escribir directamente en código binario. Los programas que traducen código simbólico al lenguaje de máquina se llaman ensambladores ("assembler", en inglés), porque son capaces de ensamblar el programa traducido a partir de varias piezas, procedimientos o subrutinas.

► Lenguaje de programación de bajo nivel. Es aquel que es fácilmente trasladado a lenguaje de máquina en el cual las instrucciones son una lista de caracteres escritos en el sistema hexadecimal. En general se utiliza este tipo de lenguaje para programar controladores (drivers).

► Lenguaje de programación de alto nivel. Se aproxima a expresiones usadas en el lenguaje habitual generalmente inglés, necesita de un traductor o intérprete para que lo entienda el ordenador.

- Ventajas del Lenguaje de Alto Nivel.
 - Mayor adaptación al equipo.
 - Posibilidad de obtener la máxima velocidad con mínimo uso de memoria.
- Desventajas del Lenguaje de Alto Nivel.
 - Imposibilidad de escribir código independiente de la máquina.
 - Mayor dificultad en la programación y en la comprensión de los programas.

- Ejemplos lenguajes de alto nivel: Ada ,ALGOL ,Visual Basic ,C ,C++ ,C# ,Clipper ,Cobol ,Fortran ,Java Lexico, Logo ,Object Pascal ,Pascal ,Perl ,PHP ,PL/SQL ,Python ,Modula-2.

► **Compilador.** Programa capaz de trasladar la instrucciones en lenguaje de programación a lenguaje de maquina. Acepta programas escritos en un lenguaje de alto nivel y los traduce a otro lenguaje, generando un programa equivalente independiente, que puede ejecutarse tantas veces como se quiera. Este proceso de traducción se conoce como compilación.

- Diferentes lenguajes en el compilador
 - El de los programas de partida (LA)
 - El de los programas equivalentes traducidos (LB), normalmente el lenguaje de máquina
 - El lenguaje en que está escrito el propio compilador (LC), que puede ser igual o diferente a LA. Aumenta la portabilidad del compilador si está escrito en el mismo lenguaje, es decir, se puede compilar a sí mismo.

- Partes de un Compilador:

- **Front End:** es la parte que analiza el código fuente, comprueba su validez, genera el árbol de derivación y rellena los valores de la tabla de símbolos. Esta parte suele ser independiente de la plataforma o sistema para el cual se vaya a compilar.
- **Back End:** es la parte que genera el código máquina, específico de una plataforma, a partir de los resultados de la fase de análisis, realizada por el *Front End*.

Esta división permite que el mismo *Back End* se utilice para generar el código máquina de varios lenguajes de programación distintos y que el mismo *Front End* que sirve para analizar el código fuente de un lenguaje de programación concreto sirva para la generación de código máquina en varias plataformas distintas. El código que genera el *Back End* normalmente no se puede ejecutar directamente, sino que necesita ser enlazado por un programa enlazador (*linker*).

- **Tipos de Compiladores.** Esta taxonomía de los tipos de compiladores no es excluyente, por lo que puede haber compiladores que se adscriban a varias categorías:

- Compiladores cruzados: generan código para un sistema distinto del que están funcionando.
- Compiladores optimizadores: realizan cambios en el código para mejorar su eficiencia, pero manteniendo la funcionalidad del programa original.
- Compiladores de una sola pasada: generan el código máquina a partir de una única lectura del código fuente.
- Compiladores de varias pasadas: necesitan leer el código fuente varias veces antes de poder producir el código máquina.
- Compiladores JIT (Just In Time): forman parte de un intérprete y compilan partes del código según se necesitan.

► **Ensambladores.** Son programas que procesan los enunciados del programa origen en lenguaje ensamblador y los traducen en archivos en lenguaje máquina que son ejecutados por un microprocesador o un microcontrolador. Los ensambladores permiten que los programas origen se escriban y se editen en una computadora para generar un código ejecutable en otra computadora. El archivo en lenguaje objeto ejecutable resultante se carga y se ejecuta en el sistema destino.

► **Lenguaje Ensamblador.** El lenguaje simbólico que se utiliza para codificar los programas origen que se procesan por el ensamblador es llamado lenguaje ensamblador. Este lenguaje es una colección de símbolos mnemónicos que representan: operaciones (mnemónicos de instrucciones para la máquina o de directrices para el ensamblador), nombres simbólicos, operadores y símbolos especiales. El lenguaje ensamblador proporciona códigos de operación de los mnemónicos para todas las instrucciones de la máquina contenidas en la lista de instrucciones. Además, el lenguaje ensamblador contiene mnemónicos directrices, los cuales especifican acciones auxiliares que se llevan a cabo por el ensamblador.

► **Ventajas y Desventajas del Lenguaje Ensamblador , respecto a un lenguaje de alto nivel:**

Ventajas del Lenguaje Ensamblador:

- Velocidad
- Eficiencia de tamaño
- Flexibilidad

Desventajas del Lenguaje Ensamblador:

- Tiempo de programación
- Programas fuente grandes
- Peligro de afectar recursos inesperadamente
- Falta de portabilidad

► Java. es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel como punteros.

El lenguaje Java se creó con cinco objetivos principales:

- Debería usar la metodología de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

El diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho de Java uno de los lenguajes con mayor crecimiento y amplitud de uso en los distintos ámbitos de la industria de la informática.

○ Entornos de funcionamiento del lenguaje java:

- Dispositivos móviles. Desde la creación de la especificación J2ME (Java 2 Platform, Micro Edition), una versión del entorno de ejecución Java reducido y altamente optimizado, especialmente desarrollado para el mercado de dispositivos electrónicos de consumo se ha producido toda una revolución en lo que a la extensión de Java se refiere. Es posible encontrar microprocesadores específicamente diseñados para ejecutar bytecode Java y software Java para tarjetas inteligentes (JavaCard), teléfonos móviles, buscapersonas, set-top-boxes, sintonizadores de TV y otros pequeños electrodomésticos. El modelo de desarrollo de estas aplicaciones es muy semejante a las applets de los navegadores salvo que en este caso se denominan MIDlets.

- En la web. Desde la primera versión de java existe la posibilidad de desarrollar pequeñas aplicaciones (Applets) en Java que luego pueden ser incrustadas en una página HTML para que sean descargadas y ejecutadas por el navegador web. Estas mini-aplicaciones se ejecutan en una JVM que el navegador tiene configurada como extensión (plug-in) en un contexto de seguridad restringido configurable para impedir la ejecución local de código potencialmente malicioso.

- Sistemas de servidor. En la parte del servidor, Java es más popular que nunca, desde la aparición de la especificación de Servlets y JSP (Java Server Pages). Hasta entonces, las aplicaciones web dinámicas de servidor que existían se basaban fundamentalmente en componentes CGI y lenguajes interpretados. Ambos tenían diversos inconvenientes (fundamentalmente lentitud, elevada carga computacional o de memoria y propensión a errores por su interpretación dinámica).

- Aplicaciones de escritorio. Hoy en día existen multitud de aplicaciones gráficas de usuario basadas en Java. El entorno de ejecución Java (JRE) se ha convertido en un componente habitual en los PCs de usuario de los sistemas operativos más usados en el mundo. Además, muchas aplicaciones Java lo incluyen dentro del propio paquete de la aplicación de modo que su ejecución en cualquier PC. En las primeras versiones de la plataforma Java existían importantes limitaciones en las APIs de desarrollo gráfico (AWT). Desde la aparición de la librería Swing la situación mejoró substancialmente y posteriormente con la aparición de librerías como SWT hacen que el desarrollo de aplicaciones de escritorio complejas y con gran dinamismo, usabilidad, etc. sea relativamente sencillo.

- Plataformas Soportadas. Una versión del entorno de ejecución Java JRE (Java Runtime Environment) está disponible en la mayoría de equipos de escritorio. Sin embargo, Microsoft no lo ha incluido por defecto en sus sistemas operativos. En el caso de Apple, éste incluye una versión propia del JRE en su sistema operativo, el Mac OS. También es un producto que por defecto aparece en la mayoría de las distribuciones de Linux. Debido a incompatibilidades entre distintas versiones del JRE, muchas aplicaciones prefieren instalar su propia copia del JRE antes que confiar su suerte a la aplicación instalada por defecto. Los desarrolladores de applets de Java o bien deben insistir a los usuarios en la actualización del JRE, o bien desarrollar bajo una versión antigua de Java y verificar el correcto funcionamiento en las versiones posteriores.

- Visual basic. Visual basic es un lenguaje de programación desarrollado por Alan Cooper para Microsoft. El lenguaje de programación es un dialecto de BASIC, con importantes añadidos. Su primera versión fue presentada en 1991 con

la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta medida también la programación misma. Visual basic ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código (programa donde se escribe el código fuente), un depurador (programa que corrige errores en el código fuente para que pueda ser compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina), y un constructor de interfaz gráfica o GUI.

- Características generales. Es un lenguaje de fácil aprendizaje pensado tanto para programadores principiantes como expertos, guiado por eventos, y centrado en un motor de formularios que facilita el rápido desarrollo de aplicaciones gráficas. Su sintaxis, derivada del antiguo BASIC, ha sido ampliada con el tiempo al agregarse las características típicas de los lenguajes estructurados modernos. Se ha agregado una implementación limitada de la programación orientada a objetos (los propios formularios y controles son objetos), aunque sí admite el polimorfismo mediante el uso de los Interfaces, no admite la herencia. No requiere de manejo de punteros y posee un manejo muy sencillo de cadenas de caracteres. Posee varias bibliotecas para manejo de bases de datos, pudiendo conectar con cualquier base de datos a través de ODBC (Informix, DBase, Access, MySQL, SQL Server, PostgreSQL ,etc) a través de ADO.

Es utilizado principalmente para aplicaciones de gestión de empresas, debido a la rapidez con la que puede hacerse un programa que utilice una base de datos sencilla, además de la abundancia de programadores en este lenguaje. El compilador de Microsoft genera ejecutables que requieren una DLL para que funcionen, en algunos casos llamada MSVBVMxy.DLL (acrónimo de "MicroSoft Visual Basic Virtual Machine x.y", siendo x.y la versión) y en otros VBRUNXXX.DLL ("Visual Basic Runtime X.XX"), que provee todas las funciones implementadas en el lenguaje. Además existen un gran número de bibliotecas (DLL) que facilitan el acceso a muchas funciones del sistema operativo y la integración con otras aplicaciones. Sin embargo esto sólo es una limitación en sistemas obsoletos, ya que las bibliotecas necesarias para ejecutar programas en Visual Basic vienen de serie en todas las versiones de Windows desde Windows 2000.

- Ventajas de Visual Basic
 - La facilidad del lenguaje permite crear aplicaciones para Windows en muy poco tiempo. En otras palabras, permite un desarrollo eficaz y menor inversión en tiempo que con otros lenguajes.
 - Permite generar librerías dinámicas (DLL) ActiveX de forma nativa y Win32

(no ActiveX, sin interfaz COM) mediante una reconfiguración de su enlazador en el proceso de compilación.

- Permite la utilización de formularios (*Forms*) tanto a partir de recursos (como en otros lenguajes) como utilizando un IDE para diseñarlos.
 - Posibilidad de desarrollar y ejecutar aplicaciones de Visual Basic 6.0 en Windows Vista sin realizar cambios en la mayoría de los casos pero no se logra aprovechar al máximo las características de este sistema como permite hacerlo Visual Basic 2005 o el próximo Visual Basic 9 de Visual Studio Orcas (Noviembre 2007).
- Desventajas de Visual Basic
- Es software propietario por parte de Microsoft, por tanto nadie que no sea del equipo de desarrollo de esta compañía decide la evolución del lenguaje.
 - En Visual Basic 6.0 y anteriores sólo existe un compilador e IDE, llamado igual que el lenguaje.
 - No existe forma alguna de exportar el código a otras plataformas fuera de Windows.
 - Los ejecutables generados son relativamente lentos en Visual Basic 6.0 y anteriores al ser código pseudo-interpretado.
 - No permite programación a bajo nivel ni incrustar secciones de código en ASM.
 - Sólo permite el uso de funciones de librerías dinámicas (DLL) stdcall.
 - Fuerte dependencia de librerías y componentes en las versiones 6.0 y anteriores, lo que dificultaba la distribución de los desarrollos entre máquinas.
 - Algunas funciones están indocumentadas.
 - El manejo de errores que tiene mediante la orden on error no sigue los patrones estructurados.
 - No incluye operadores de desplazamiento de bits como parte del lenguaje.
 - No avisa de ciertos errores o advertencias (se puede configurar el compilador para generar ejecutables sin los controladores de desbordamiento de enteros o las comprobaciones de límites en matrices entre otros, dejando así más de la mano del programador la tarea de controlar dichos errores).

1.2.8. Base de datos. Una base de datos o banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

► Base de datos según la variabilidad de los datos almacenados.

- Bases de datos estáticas. Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

- Bases de datos dinámicas. Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc.

► Modelos de bases de datos. Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos. Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

- Bases de datos jerárquicas. Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

- Base de datos de red. Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la

dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

- Base de datos relacional. Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información. El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

- Bases de datos orientadas a objetos. Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz.

Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

- Bases de datos documentales. Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes.

- Base de datos deductivas. Un sistema de base de datos Deductivas, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática.

- Gestión de bases de datos distribuida. La base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a distintas universidades, sucursales de tiendas, etc.

1.2.9 Redes de computadores. La definición más clara de una red es la de un sistema de comunicaciones, ya que permite comunicarse con otros usuarios y compartir archivos y periféricos. Es decir es un sistema de comunicaciones que conecta a varias unidades y que les permite intercambiar información. Se entiende por red al conjunto interconectado de ordenadores autónomos. Se dice que dos ordenadores están interconectados, si éstos son capaces de intercambiar información. La conexión no necesita hacerse a través de un hilo de cobre, también puede hacerse mediante el uso de láser, microondas y satélites de comunicación. Son muchas las organizaciones que cuentan con un número considerable de ordenadores en operación y con frecuencia alejados unos de otros. Por ejemplo, una compañía con varias fábricas puede tener un ordenador en cada una de ellas para mantener un seguimiento de inventarios, observar la productividad y llevar la nómina local. Inicialmente cada uno de estos ordenadores puede haber estado trabajando en forma aislada de las demás pero, en algún momento, la administración puede decidir interconectarlos para tener así la capacidad de extraer y correlacionar información referente a toda la compañía. Es decir el objetivo básico es compartir recursos, es decir hacer que todos los programas, datos y equipos estén disponibles para cualquiera de la red que lo solicite, sin importar la localización del recurso y del usuario.

Un segundo objetivo es proporcionar una alta fiabilidad, al contar con fuentes alternativas de suministro. Todos los archivos podrían duplicarse en dos o tres máquinas, de tal manera que si una no se encuentra disponibles, podría utilizarse algunas de las copias.

► Diferencia entre sistemas distribuidos y sistemas centralizados. En los entornos con grandes computadoras y mini computadoras, el procesamiento y la memoria se encuentran centralizados. Hay varias razones para ello, incluyendo el costo, la seguridad y la gestión. La computadora central se convierte en el núcleo de la organización de proceso de datos, habiendo un equipo de profesionales que tienen como única tarea el trabajar y administrar el sistema. Los terminales conectados al ordenador central permiten que otros usuarios puedan compartir las posibilidades de cálculo y la memoria de las computadoras centrales.

Este tipo de proceso centralizado se diferencia del sistema de proceso distribuido utilizado por las LAN. En un sistema de proceso distribuido, la mayor parte de los procesos se lleva a cabo en la memoria individual de las computadoras personales, a las que denomina estaciones de trabajo. El servidor de archivos o sistema central se convierte en un lugar para almacenar los archivos y para gestionar la red, además de ser el lugar al que se conectan las impresoras y otros recursos compartidos.

► Cableado. Una vez que se tienen las estaciones de trabajo, el servidor y las placas de red, se requiere interconectar todo el conjunto. El tipo de cable utilizado depende de muchos factores, que se mencionarán a continuación:

Los tipos de cableado de red más populares son: par trenzado, cable coaxial y fibra óptica. Además se pueden realizar conexiones a través de radio o microondas. Cada tipo de cable o método tiene sus ventajas y desventajas. Algunos son propensos a interferencias, mientras otros no pueden usarse por razones de seguridad. La velocidad y longitud del tendido son otros factores a tener en cuenta el tipo de cable a utilizar.

○ Par trenzado. Consiste en dos hilos de cobre trenzado, aislados de forma independiente y trenzados entre sí. El par está cubierto por una capa aislante externa tiene un ancho de banda de 3MHZ, y la separación entre repetidores para evitar pérdidas se encuentra de 2 a 10 kilómetros evitando pérdidas a menores distancias. Entre sus principales ventajas están:

- Es una tecnología bien estudiada
- No requiere una habilidad especial para instalación
- La instalación es rápida y fácil
- La emisión de señales al exterior es mínima.
- Ofrece alguna inmunidad frente a interferencias, modulación cruzada y corrosión.

- Cable coaxial. Se compone de un hilo conductor de cobre envuelto por una malla trenzada plana que hace las funciones de tierra. Entre el hilo conductor y la malla hay una capa gruesa de material aislante, y todo el conjunto está protegido por una cobertura externa. El cable está disponible en dos espesores: grueso y fino. El cable grueso soporta largas distancias, pero es más caro. El cable fino puede ser más práctico para conectar puntos cercanos.

El Cable Coaxial Ofrece las Siguietes Ventajas

- Soporta comunicaciones en banda ancha y en banda base.
- Es útil para varias señales, incluyendo voz, video y datos.
- Es una tecnología bien estudiada.

- Conexión fibra optica. Esta conexión es cara, permite transmitir la información a gran velocidad e impide la intervención de las líneas. Como la señal es transmitida a través de luz, existen muy pocas posibilidades de interferencias eléctricas o emisión de señal. El cable consta de dos núcleos ópticos, uno interno y otro externo, que refractan la luz de forma distinta. La fibra está encapsulada en un cable protector.

Ventajas de la fibra optica:

- Alta velocidad de transmisión
- No emite señales eléctricas o magnéticas, lo cual redundo en la seguridad
- Inmunidad frente a interferencias y modulación cruzada.
- Mayor economía que el cable coaxial en algunas instalaciones.
- Soporta mayores distancias

► Redes según la cobertura del servicio.

- Redes de area local (LAN). Son redes que interconectan equipos dentro de un entorno físico reducido. En general no se extiende más allá de un edificio, recinto o campús.
- Redes de area extensa (WAN). Son las que unen equipos instalados en distintos edificios e inclusive en distintas ciudades. Utilizan normalmente enlaces de telecomunicación de la compañía telefónica.

- Distribución y topología de redes. Topología de red es la forma en que se distribuyen los cables de la red para conectarse con el servidor y con cada una de las estaciones de trabajo. La topología de una red es similar a un plano de la red dibujado en un papel, ya que se pueden tender cables a cada estación de trabajo y servidor de la red. La topología determina donde pueden colocarse las

estaciones de trabajo, la facilidad con que se tenderá el cable y el corte de todo el sistema de cableado. La flexibilidad de una red en cuanto a sus necesidades futuras se refiere, depende en gran parte de la topología establecida.

- Topología estrella. Se utiliza un dispositivo como punto de conexión de todos los cables que parten de las estaciones de trabajo. El dispositivo central puede ser el servidor de archivos en sí o un dispositivo especial de conexión. El diagnóstico de problemas es fácil, debido a que las estaciones de trabajo se comunican a través del equipo central. Los fallos en el nodo central son fáciles de detectar y es fácil cambiar los cables. La colisión entre datos es imposible, ya que cada estación tiene su propio cable, y resulta fácil ampliar el sistema. En algunas empresas tienden a agruparse los cables en la unidad central lo cual puede ocasionar errores de gestión.

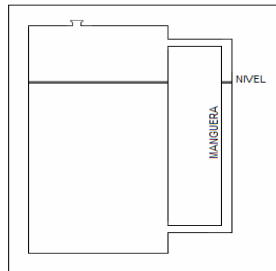
- Topología bus. El servidor y todas las estaciones están conectados a un cable general central. Todos los nodos comparten este cable y éste necesita acopladores en ambos extremos. Las señales y los datos van y vienen por el cable, asociados a una dirección destino. Cada nodo verifica las direcciones de los paquetes que circulan por la red para ver si alguna coincide con la suya propia. El cable puede extenderse de cualquier forma por las paredes y techos de la instalación. Ej: Ethernet y G-Net. La topología bus usa una cantidad mínima de cable y el cable es muy fácil de instalar, ya que puede extenderse por un edificio en las mejores rutas posibles. Así el cable debe ir de equipo en equipo. Las principales desventajas de esta topología son: El cable central puede convertirse en un cuello de botella en entornos con un tráfico elevado, ya que todas las estaciones de trabajo comparten el mismo cable. Es difícil aislar los problemas de cableado en la red y determinar que estación o segmento de cable los origina, ya que todas las estaciones están en el mismo cable. Una rotura de cable hará caer el sistema.

- Topología anillo. Las señales viajan en una única dirección a lo largo del cable en forma de un bucle cerrado. En cada momento, cada nodo pasa las señales a otro nodo. Con la topología en anillo, las redes pueden extenderse a menudo a largas distancias, y el coste total del cableado será menor que en una configuración en estrella y casi igual a la bus. Una rotura del cable hará caer el sistema. Actualmente existen sistemas alternativos que evitan que esto ocurra.

1.3. ANTECEDENTES

En el momento actual, las plantas eléctricas de la central telefónica de EMCALI, cuentan con un sistema de medición de nivel que es muy similar al Medidor de Sonda, el cual consiste en una manguera la cual esta conectada al tanque, en la parte inferior y parte superior (Ver Figura 17).

Figura 17. Medidor de sonda EMCALI .



Fuente: BELTRÁN PROVOSTE, Cristhian. Medición de nivel [en línea]. Barcelona: infoPLC.net, 2008. Modificada por última vez 13 septiembre 2007. [Consultado 03 Marzo de 2007]. Disponible Internet: http://www.infoplcn.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.pdf

La única diferencia es que esta no cuenta con la regla graduada, la cual nos indica en que nivel se encuentra el tanque de combustible, como se puede ver este sistema es muy efectivo y confiable, pero tiene un inconveniente y es que por el tipo de combustible utilizado (ACPM) la visibilidad de la manguera con el tiempo, se hace un poco dificultosa y en algunos casos obligando al operario a utilizar métodos poco convencionales como es, la utilización de un pedazo de madera la cual es introducida por la parte posterior del tanque, y de esta forma poder saber en que nivel se encuentra. Por otro lado este sistema implica que la persona encargada de medir el nivel se tenga que desplazar hasta donde se encuentra cada uno de los tanques, registrar el nivel en que se encuentra, presentar el informe y luego regresar para hacer el reabastecimiento de combustible.

1.4. OBJETIVO GENERAL

Diseñar el prototipo de un dispositivo que permita monitorear de forma centralizada, la cantidad de combustible que hay en los tanques de almacenamiento de Las Plantas Eléctricas de Las Centrales Telefónicas de EMCALI

1.5. OBJETIVOS ESPECIFICOS

- Conocer el funcionamiento y el estado de Las Planta Eléctricas de La Central Telefónica de EMCALI.
- Desarrollar el dispositivo encargado de procesar la información registrada por el sensor la cantidad de combustible que hay en uno de los tanques de una de las plantas Eléctricas de La Central Telefónica de EMCALI, para su posterior visualización, monitoreo y almacenamiento.
- Desarrollar la interfaz grafica del programa encargado de monitorear, almacenar la información y conectar las centrales secundarias con la central de Colon.
- Realizar pruebas y ajustes de la comunicación y de la interfaz grafica.

1.6. JUSTIFICACION

Las centrales telefónicas de EMCALI, cuentan con dispositivos de apoyo que garantizan que el servicio que presta esta empresa no sea interrumpido cuando se presente algún problema en las salas de conmutación, uno de los muchos inconvenientes que se presentan son el corte del fluido eléctrico, el cual es contrarestado con la utilización de plantas eléctricas las cuales funcionan en el momento que se presenta la emergencia (corte del fluido eléctrico), estas plantas funcionan con combustible (ACPM), y cuentan con unos tanque de almacenamiento los cuales son revisados diariamente para verificar en que nivel se encuentra el combustible y de esta forma poder reabastecerlos. Este procedimiento se hace de forma presencial en cada una de las plantas eléctricas que hay en las centrales telefónicas de EMCALI, este procedimiento se realiza de la siguiente manera una persona se encarga de pasar revista en cada una de las plantas eléctricas que hay en una de las centrales, luego de pasar revista esta persona hace el reporte y si hay que reabastecer de combustible debe desplazarse hasta donde se encuentra el combustible y traerlo y de esta forma hacer el reabastecimiento, como se puede observar es un procedimiento que demanda tiempo si se tiene en cuenta que en algunas centrales telefónicas hay mas de 50 plantas eléctricas, por esta razón la importancia de la realización de este proyecto el cual permitirá que una sola persona se encargue de realizar la supervisión del nivel en que se encuentra los tanque de almacenamiento de combustible de las P.E. sin tener que desplazarse, disminuyendo los costos que implica este procedimiento, dado que una persona se encarga de la supervisión y otra del reabastecimiento y solo seria necesario el desplazamiento de la persona encargada de reabastecer las P.E.

1.7. METODOLOGIA

Para desarrollo del proyecto se dividirá en 4 etapas básicas que propondrán un mayor enfoque en aspectos técnicos necesarios para la implementación del sistema.

Reconocimiento

Conocer el funcionamiento y el estado de Las Planta Eléctricas de La Central Telefónica de EMCALI.

Investigar aspectos técnicos de las posibles opciones para el desarrollo del proyecto.

Estudio y Análisis

Estudio, análisis y selección de los dispositivos, tecnologías mas adecuadas para el desarrollo del sistema a diseñar.

Etapas de diseño

Desarrollar el dispositivo encargado de procesar la información registrada por el sensor, para su posterior visualización, monitoreo y almacenamiento.

Desarrollar la interfaz grafica del programa encargado de monitorear, almacenar la información y conectar las centrales secundarias con la central de Colon.

Etapas de Realización de Pruebas

Realizar pruebas y ajustes de la comunicación y de la interfaz grafica.

Elaboración del documento que contiene la descripción del sistema diseñado, el manual de usuario y las conclusiones obtenidas en el desarrollo del proyecto.

1.8. CRONOGRAMA

Tabla 5. Cronograma de trabajo

	JUNIO				JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE			
SEMANA	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ETAPA 1																								
ETAPA 2																								
ETAPA 3																								
ETAPA 4																								

2. DISEÑO CONCEPTUAL

2.1. ESTUDIO Y ANÁLISIS

En este capítulo se desarrollaran el estudio, análisis y selección de los dispositivos, tecnologías mas adecuadas para el desarrollo del sistema a diseñar. Correspondiente a la segunda etapa del desarrollo del proyecto. El estudio y análisis se realizaran de acuerdo a los métodos de diseño electrónico basadosg en la identificación, ponderación, caracterización de los requerimientos principales del sistema y sus funciones.

2.1.1. Identificación de los requerimientos del sistema. Requerimientos del sistema. Para el desarrollo o diseño conceptual del sistema de monitoreo se tuvieron en cuenta los siguientes requerimientos o características que se tendrán en cuenta al momento del diseño.

► Requerimientos del diseño

- Fácil de manejar
- De bajo costo
- Fácil de instalar
- Transportable
- Consumo de potencia mínimo
- Con posibilidad para ampliaciones o actualizaciones
- Evitar riesgos en la medición de nivel del combustible
- Fácil acceso a la información registrada durante el día

► Desempeño funcional

- Desempeño eficiente
- Visualización llamativa
- Visualización dinámica de las variables monitoreadas

► Restricciones espaciales

El sistema de monitoreo debe ocupar el menor espacio posible de acuerdo al lugar de instalación. Por ejemplo un cuarto, en cercanías a un tanque de combustible.

► Apariencia

- Que el espacio que ocupado por dispositivo sea mínimo.
- Que pueda ser ubicado en cualquier lugar sin que llame la atención.

► Tiempo

- Instalación rápida máximo 3 horas

► Costo

- Mantenimiento y repuestos son de bajo costo

2.1.2. Clasificación de los requerimientos del sistema. A cada requerimiento se le atribuye un valor cuantificado de uno a cinco (1 a 5) el cual determina el nivel de importancia de cada requerimiento al momento de realizar el diseño y desarrollo del sistema (Ver Tabla 6).

Tabla 6. Clasificación de los requerimientos del sistema.

	REQUERIMIENTOS DEL SISTEMA	IMPORTANCIA
1	Fácil de manejar	4
2	De bajo costo	5
3	Fácil de instalar	5
4	Transportable	4
5	Consumo de potencia mínimo	5
6	Con posibilidad para ampliaciones o actualizaciones	4
7	Evitar riesgos en la medición de nivel del combustible	4
8	Fácil acceso a la información registrada	4
9	Desempeño eficiente	4
10	Visualización llamativa	3
11	Visualización dinámica de las variables monitoreadas.	4
12	Ocupar el menor espacio posible	4
13	Instalación rápida	4

- Jerarquización de los requerimientos. Los requerimientos del cliente de forma cualitativa, de acuerdo a su grado de importancia para determinar la eficacia o intensidad con la que se debe enfocar en cada uno de estos (Ver Tabla7).

Tabla 7. Jerarquización de los requerimientos

No.	Requerimiento del Sistema	NECESIDAD		
		Primaria	Secundaria	Terciaria
1	Fácil de manejar			
2	De bajo costo			
3	Fácil de instalar			
4	Transportable			
5	Consumo de potencia mínimo			
6	Ampliaciones o actualizaciones			
7	Evitar riesgos en la medición			
8	Acceso a la información registrada			
9	Desempeño eficiente			
10	Visualización llamativa			
11	Visualización dinámica			
12	Ocupar el menor espacio posible			
13	Instalación rápida			

- Métricas y relación con los requerimientos. Para cuantificar y expresar las necesidades en un lenguaje de ingeniería con el objetivo de establecer las especificaciones preliminares del producto (Ver Tabla 8).

Tabla 8. Métricas y unidades de medida que determinan al dispositivo

No.	Requerimiento	Métrica	Unidad
1	7,9	Medida de Nivel	M, Cm
2	4,5	Tipo de alimentación	Vd, Vac
3	5	Potencia consumida	Watts
4	4	Peso	Kg
5	4,12	Dimensiones	Cm
6	9	Velocidad de comunicación	Bytes/seg
7	4,6,12	Área de trabajo	M, Cm
8	1,8,9	Manejo base de datos	Alto, Medio, Bajo
9	2,6	Costo del sistema	Pesos
10	1,6,9,13	Tiempo instalación	Minutos, Horas
11	7,9	Nivel de confiabilidad	Alto, Medio, Bajo
12	1,3,13	Dependencia del usuario	Alto, Medio, Bajo

Para identificar la relevancia de cada una de las métricas y su relación con los requerimientos se realizó una matriz donde se determina el nivel de prioridad de las métricas, con respecto a los requerimientos (Ver Tablas 9, 10).

Tabla 9. Metricas

No.	Métrica	Unidad
1	Medida de Nivel	M, Cm
2	Tipo de alimentación	Vd, Vac
3	Potencia consumida	Watts
4	Peso	Kg
5	Dimensiones	Cm
6	Velocidad de comunicación	Bytes/seg
7	Área de trabajo	M, Cm
8	Manejo base de datos	Alto, Medio, Bajo
9	Costo del sistema	Pesos
10	Tiempo instalación	Minutos, Horas
11	Nivel de confiabilidad	Alto,Medio, Bajo
12	Dependencia del usuario	Alto, Medio, Bajo

Tabla 10. Matriz métricas y relación con los requerimientos

MATRIZ METRICAS Y RELACIÓN CON LOS REQUERIMIENTOS			METRICAS												Total
No.	REQUERIMIENTOS	Prioridad	1	2	3	4	5	6	7	8	9	10	11	12	
1	Fácil de manejar	4								Θ		Θ		Ω	60
2	De bajo costo	5									Θ				15
3	Fácil de instalar	5												Ω	45
4	Transportable	4		Ω		Ω	Ω		Ω						144
5	Consumo de potencia mínimo	5		Ω	Θ										60
6	Ampliaciones o actualizaciones	4							Ω		Θ	Θ			60
7	Evitar riesgos en la medición	4	Θ										Θ		72
8	Fácil acceso a la información	4								Θ					36
9	Desempeño eficiente	4	Θ					Ω		Θ		Θ	Θ		156
12	Ocupar el menor espacio posible	4					Θ		Ω						48
13	Instalación rápida	4										Θ	Θ		72

Nivel de prioridad 0

Δ Nivel de prioridad 1

Θ Nivel de prioridad 3

Ω Nivel de prioridad 9

- De este análisis por relación se concluye que las características más importantes son la transportabilidad del sistema, su desempeño eficiente, instalación rápida y evitar riesgos en la medición.
- Las características de menor importancia son el fácil acceso a la información y el costo del producto.
- Las características de con un segundo lugar de importancia son, fácil de manejar, fácil de instalar, consumo de potencia mínimo, ocupar el menor espacio posible.

De acuerdo a los resultados se procederá a realizar el diseño teniendo en cuenta el grado de importancia de los requerimientos y su clasificación con respecto a las métricas, relación anteriormente analizada.

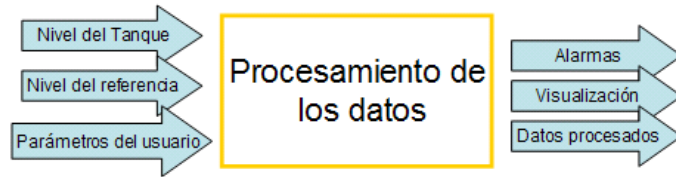
2.1.3. Generación de alternativas de solución. En el proceso de generación de alternativas de solución se tuvo en cuenta el resultado del estudio y el análisis de los requerimientos del sistema con su respectivo grado de importancia y los conceptos especificados en el marco teórico.

► Descomposición funcional. En la descomposición funcional se representa los diferentes procesos de análisis y transformación que debe llevar a cabo el dispositivo con las entradas y sus respectivas salidas.

Los diferentes procesos o funciones representados a continuación se plantearon teniendo en cuenta las principales características que conforman un sistema SCADA.

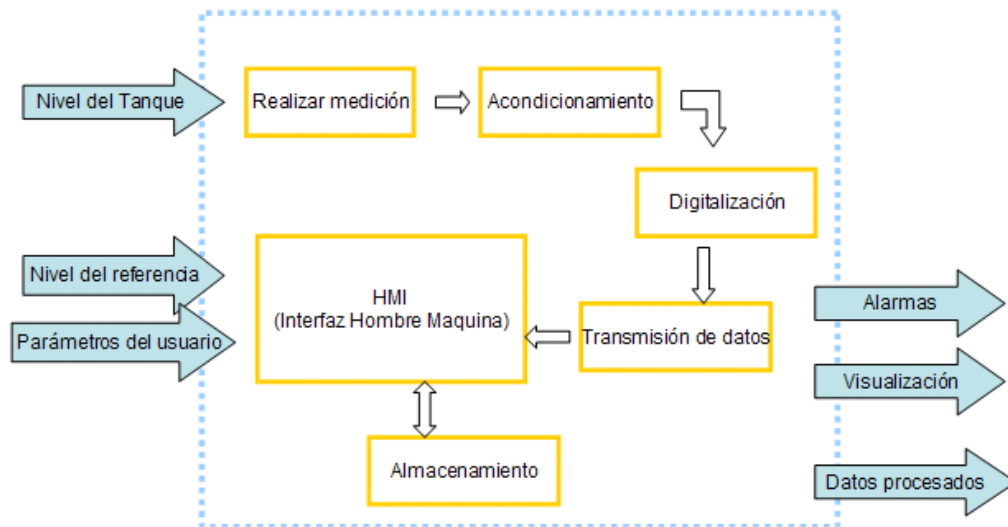
Para la representación grafica del dispositivo conformado por entradas y salidas se planteó un diagrama de bloques que hace referencia a la caja negra del dispositivo y las diferentes funciones que se ejecutan dentro de de la caja negra. (Ver Figuras 18, 19).

Figura 18. Caja negra del sistema



En la caja negra el bloque de procesamiento de datos es el encargado de realizar la medición del nivel actual en el tanque, el acondicionamiento de estos datos y la transmisión de esta información a la HMI (interfaz hombre maquina) donde de acuerdo a los parámetros introducidos por el usuario se visualizara, almacenara y analizara.

Figura 19. Descomposición funcional del sistema



- Función medición. En este bloque se realizara la medición del nivel actual de combustible en cada uno de los tanques conectados al sistema SCADA, la medición se realizara con un transductor en este caso un sensor de nivel que de acuerdo a su rango de medición permitirá transformar la variable física en una analoga para su posterior digitalización y acondicionamiento.

- Función acondicionamiento. Este bloque se encargara de acondicionar la señal, convertirá la señal de salida de los elementos primarios (transductores) de medida en señales normalizadas para su posterior digitalización o transmisión al centro de control o monitoreo. El acondicionamiento de señal se realiza cuando la señal entregada por un transductor o sensor es de muy bajo nivel, generalmente requiere amplificación u otro tipo de acondicionamiento, antes de procesarla. Esto puede incluir filtrado análogo o filtrado digital, amplificación, separación o aislamiento, linealización, decodificación o detección de la señal.

Las señales normalizadas son:

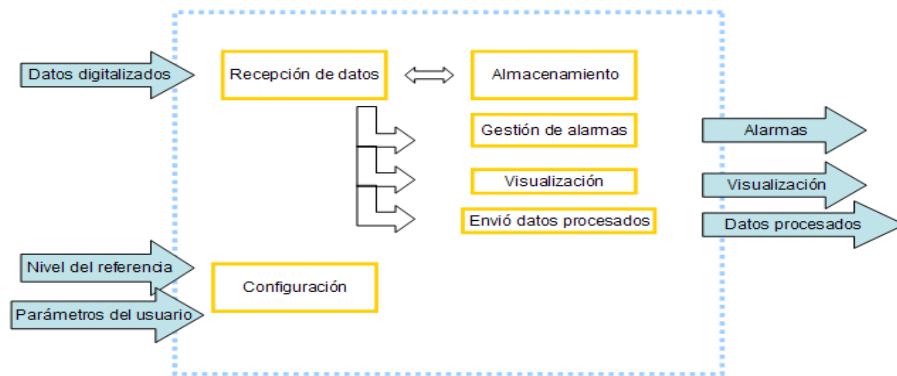
- Corriente: 4mA-20mA.
- Voltaje: TTL, 0V-5V y 0V-10V.
- Neumática: 3 PSI - 15 PSI.

- Función digitalización. En este bloque se llevara a cabo la conversión A/D (análogo/digital) esta conversión es necesaria cuando se debe expresar en forma digital una serie de magnitudes medidas, para procesarlas en una computadora o microcontrolador, visualizarlas, procesarlas y almacenarlas. Algunas de las características más importantes de este proceso son el tiempo de conversión y el mínimo cambio en la señal que puede ser observado en nuestro instrumento también llamado la resolución del sistema de conversión.

- Función transmisión de datos. Este bloque es el encargado de transmitir la información anteriormente digitalizada. Normalmente la transmisión se realiza de forma sincrónica para evitar perdidas en la información y para facilitar su posterior procesamiento.

- Función HMI. En este bloque se llevara a cabo todos los procesos donde tenga que intervenir o interactuar el usuario con la información y el procesamiento de la misma, también se debe encargara de permitir la configuración parcial del sistema, indicar el estado del proceso, indicar situaciones de alarma (Ver Figura 20).

Figura 20. Función HMI



- Sub-función configuración. Esta sub-función se encargara de guardar los parámetros introducidos por el usuario, y determinar el comportamiento de la interfaz hombre maquina de acuerdo a a estos parámetros.
- Sub-función recepción de datos. En esta subfunción se encargara de recibir los datos digitalizados, organizarlos según su punto de origen.³
- Sub-función almacenamiento. En esta subfunción se desarrollara el proceso de almacenamiento de los datos entrantes en una base datos de acuerdo a su punto de origen y características.
- Sub-función gestión de alarmas. En esta subfunción se generaran los eventos de alarma y se visualizaran de forma llamativa para que el usuario se encargue de atender el problema lo más rápido posible.
- Sub-función visualización. En esta subfunción se generaran los eventos y procesos que representan el estado actual de los tanques de combustible de acuerdo a la información recibida.
- Sub-función envió datos procesados. En esta subfunción se organizaran y enviaran los datos de una central a otra según el estado actual de la comunicación y de los datos recibidos.

2.1.4. Análisis de las alternativas de solución. De acuerdo a la descomposición funcional de todo el sistema con base en el marco teórico se generaron algunas alternativas para cada bloque y se organizaron por medio de una tabla morfológica (Ver Tabla 16).

► Función Medición

Tabla 11. Comparacion de soluciones funcion medicion.

Medio	Ventajas	Desventajas
Medidor de sonda	<ul style="list-style-type: none"> - Medicion precisa - Se usa con cualquier liquido 	<ul style="list-style-type: none"> - Realizar la medicion requiere mucho tiempo y debe realizarse directamente.
Medidor de cinta y plomada	<ul style="list-style-type: none"> - Medicion precisa - Se usa con cualquier liquido - Se usa en tanques de gran capacidad 	<ul style="list-style-type: none"> - Realizar la medicion requiere mucho tiempo y debe realizarse directamente.
Medidor de nivel de cristal	<ul style="list-style-type: none"> - Medicion precisa 	<ul style="list-style-type: none"> - Realizar la medicion requiere mucho tiempo y debe realizarse directamente.
Medidor de flotador	<ul style="list-style-type: none"> - Medicion precisa - Se usa con cualquier liquido - Se usa en tanques de gran capacidad 	<ul style="list-style-type: none"> - las partes móviles están expuestas al fluido y pueden romperse - el flotador debe mantenerse limpio
Medidor manométrico	<ul style="list-style-type: none"> - Medicion precisa 	<ul style="list-style-type: none"> - rango de medida es bastante pequeño - sólo sirve para fluidos limpios - La medida está limitada a tanques abiertos
Medidor de membrana	<ul style="list-style-type: none"> - Medicion precisa - Se usa con cualquier liquido 	<ul style="list-style-type: none"> - El instrumento es delicado. - Necesita continuo mantenimiento
Medidor de tipo burbujeo	<ul style="list-style-type: none"> - Medicion directa y precisa - Se usa con cualquier liquido 	<ul style="list-style-type: none"> - La medida está limitada a tanques abiertos
Medidor de presión diferencial	<ul style="list-style-type: none"> - Medicion precisa - el tipo más utilizado para líquidos con sólidos en suspensión, gasolina y petróleo - Realizar la medicion requiere poco tiempo 	<ul style="list-style-type: none"> - Sus salidas analogas deben acondicionarse. - En liquidos corrosivos el diafragma debe modificarse por uno anticorrosivo.
Medidor de Nivel Resistivo	<ul style="list-style-type: none"> - Campo de medida es grande - Medicion directa y precisa - No se puede usar con liquidos inflamables. 	<ul style="list-style-type: none"> - Debe tener contacto con el líquido para realizar la medicion.
Medidor de Nivel Capacitivo	<ul style="list-style-type: none"> - Presentan una buena resistencia a la corrosión - son de fácil limpieza 	<ul style="list-style-type: none"> - la temperatura puede afectar la medicion
Medidor Ultrasónico	<ul style="list-style-type: none"> - Medicion precisa - Son adecuados para todos los tipos de tanques 	<ul style="list-style-type: none"> - sensibles a la densidad de los fluidos - registra señales erróneas cuando la superficie del nivel del líquido no es nítida
Medidor de Radiación	<ul style="list-style-type: none"> - Se usa en tanque de presiones elevadas y difícil acceso o peligrosos - Medicion directa y precisa - Son adecuados para todos los tipos de tanques 	<ul style="list-style-type: none"> - Su lectura viene influida por el aire o los gases disueltos en el líquido - el sistema es caro - la instalación puede presentar peligro de contaminación radiactiva
Medidor Láser	<ul style="list-style-type: none"> - Medicion precisa - Son adecuados para todos los tipos de tanques - Se usan en fluidos a altas temperaturas. 	<ul style="list-style-type: none"> - el sistema tiene un precio elevado

Con Base en la tabla y en los requerimientos que debe cumplir el sistema de monitoreo se descartaron las soluciones con las siguientes desventajas:

- Donde la medicion debe realizarse directamente y requiere mucho tiempo.
- Los que tienen un precio elevado.
- Los que deben tener un contacto directo con el liquido.
- Los que son afectados por la temperatura, el aire o los gases disueltos en el liquido.
- Los que requieren que el tanque se encuentre abierto.

Estas soluciones se descartaron por las características del ACPM (ver Anexo 5) , la distribución de los tanques en las centrales, el costo de los sensores y la forma en la cual deben estar instalados.

Con Base en la tabla se escogieron las siguientes soluciones:

- Las que se usan en cualquier líquido.
- Tiene una medicion precisa
- Realizar la medicion requiere poco tiempo.

► Función Acondicionamiento

Tabla 12. Comparacion de soluciones funcion acondicionamiento.

Medio	Ventajas	Desventajas
AOP- Divisor Voltaje	- Acople de impedancias. - Tiene un precio muy asequible.	- No sirve para acondicionar señales de voltaje muy pequeñas
AOP- Instrumentación	- Acople de impedancias. - Acondiciona señales de voltaje del orden de los milivoltios. - Alta precision	- Tiene un precio elevado.
Puente de wheastone	- Acondiciona señales de voltaje del orden de los milivoltios.	

Con Base en la tabla y en los requerimientos se determino que la función de acondicionamiento se puede implementar con cualquiera de las soluciones planteadas.

► Función Digitalización

Tabla 13. Comparacion de soluciones funcion digitalizacion.

Medio	Ventajas	Desventajas
ADC de 8 bits	<ul style="list-style-type: none"> - Permite digitalizar un máximo de 8 señales análogas. - Alta resolución. - Tiempo de conversión. 	
ADC de 10 bits	<ul style="list-style-type: none"> - Permite digitalizar un máximo de 10 señales análogas. - Tiempo de conversión. 	
LM339	<ul style="list-style-type: none"> - Tiene un precio muy asequible. 	<ul style="list-style-type: none"> - Solo puede digitalizar simultáneamente una señal analoga.
LM555	<ul style="list-style-type: none"> - Tiene un precio muy asequible. 	<ul style="list-style-type: none"> - Solo puede digitalizar simultáneamente una señal analoga.

Con Base en la tabla y en los requerimientos que debe cumplir el sistema de monitoreo se descartaron las soluciones con las siguientes desventajas:

- Solo se puede digitalizar una señal analoga.

Estos metodos se descartaron porque para determinar el nivel de los tanques se necesita una mayor resolucion y un mayor número de bits que represente la señal analoga registrada.

► Función Transmisión de Datos

Tabla 14. Comparacion de soluciones funcion transmision de datos.

Medio	Ventajas	Desventajas
μC–Serial:	<ul style="list-style-type: none"> - Altas velocidades de transmisión. - Se adapta a cualquier dispositivo con puerto serial. - Puede transmitir en forma sincronica o asincronica. 	<ul style="list-style-type: none"> - Tiene como límite de 15 m de distancia para transmitir sin perdidas.
Paralelo:	<ul style="list-style-type: none"> - Transmite todos los bits de información al mismo tiempo. 	<ul style="list-style-type: none"> - Utiliza todo el puerto paralelo, dificultando la ampliación del sistema.
μC—USB:	<ul style="list-style-type: none"> - Altas velocidades de transmisión. 	<ul style="list-style-type: none"> - Necesita un modulo de conversión de RS232 a USB. - Precio elevado.

Con Base en la tabla y en los requerimientos que debe cumplir el sistema de monitoreo se descartaron las soluciones con las siguientes desventajas:

- Donde se dificulta la ampliación del sistema mucho tiempo.
- Los que tienen un precio elevado.

► Función HMI (interfaz hombre maquina)

Tabla 15. Comparacion de soluciones funcion HMI

Medio	Ventajas	Desventajas
C++:	<ul style="list-style-type: none"> - Altamente configurable. - Compatible con varias plataformas de desarrollo. - Basado en el manejo de librerías. - Permite programación a bajo nivel. 	<ul style="list-style-type: none"> - Se dificulta el desarrollo de interfaces graficas.
Java:	<ul style="list-style-type: none"> - Independencia de la plataforma de desarrollo - orientados a objetos 	<ul style="list-style-type: none"> - No tiene una interfaz grafica incluida. - Los ejecutables generados son relativamente lentos
Visual Basic:	<ul style="list-style-type: none"> - Altamente configurable. - Interfaz grafic incluida. - Permite generar archivos ejecutables y de instalación - basado en el manejo de librerías DLL. - Menor inversión en el tiempo de programación que con otros lenguajes 	<ul style="list-style-type: none"> - Programación orientada a objetos limitada. - sólo existe un compilador para este lenguaje. - No existe forma alguna de exportar el código a otras plataformas fuera de Windows.

Con Base en la tabla, los requerimientos y conocimientos previos de los lenguajes de programación se descartaron las soluciones con las siguientes desventajas:

- Se dificulta el desarrollo de interfaces graficas.
- No tiene una interfaz grafica incluida.

Se determino que la mejor opcion para la implementacion de la función HMI (interfaz hombre maquina) es con el lenguaje visual Basic, por el conocimiento previo que se tiene de este, la amplia gama de librerias y controles que posee, por la facilidad con la cual se desarrolla una interfaz grafica en este entorno, y especialmente cuando se refiere al manejo y comunicación de datos.

Tabla 16. Tabla morfológica

FUNCIONES	MEDIOS		
	Medidor de flotador	Medidor de membrana	Medidor de Presión
Medición			
Acondicionamiento	AOP-Divisor Voltaje	AOP-Instrumentación	Puente de wheastone
Digitalización	ADC 8 Bits	ADC 10 Bits	
Transmisión de datos	μC –Serial		
HMI	C++	Java	Visual Basic

2.1.5. Selección de la solución. Con base en el resultado del análisis y la determinación de las ventajas y desventajas de cada una de las alternativas de cada función la combinación de soluciones es la siguiente. (Ver Tablas 17, 18).

Tabla 17. Tabla morfológica selección de la solución

Funciones									
Medición		Acondicionamiento		Digitalización		Transmisión		HMI	
1	Medidor de flotador	1	AOP-Divisor Voltaje	1	ADC 8 Bits	1	μC – Serial	1	Visual Basic
2	Medidor de membrana	2	AOP-Instrumentación	2	ADC 10 Bits	2		2	
3	Medidor de Presión	3	Puente de wheastone	3		3		3	

Tabla 18. Combinación de soluciones

Combinaciones	Posible	Solucion
1-1-1-1-1	+	A
1-2-1-1-1	-	B
1-3-1-1-1	-	C
1-1-2-1-1	-	D
1-2-2-1-1	-	E
1-3-2-1-1	-	F
2-1-1-1-1	+	G
2-2-1-1-1	-	H
2-3-1-1-1	-	I
2-1-2-1-1	-	J
2-2-2-1-1	-	K
2-3-2-1-1	-	L
3-1-1-1-1	+	M
3-2-1-1-1	-	N
3-3-1-1-1	-	O
3-1-2-1-1	-	P
3-2-2-1-1	-	Q
3-3-2-1-1	-	R

► Justificación de la selección de la solución. Las soluciones marcadas con menos se descartaron por las siguientes razones:

- Alto Precio de los componentes.
- En el caso del las combinaciones con el ADC con una resolución de 10 bits se descartaron, porque no es necesario tener un sistema de conversión con tanta precisión.
- Las combinaciones con el puente de wheastone se descartaron por que en el diseño del dispositivo no se manejan niveles de voltaje en el orden de los milivoltios.

Las soluciones que se marcaron con (+) son aquellas que se adaptan mejor al diseño del dispositivo.

Solucion A. Entre las características de esta solución están:

- El sistema se puede implementar fácilmente y a un bajo costo.
- Una de sus desventajas es que en el caso del sensor las partes móviles están expuestas y pueden romperse.
- Al flotador se le debe realizar mantenimiento constante.
- Solo es necesario realizar un acople de impedancias y un divisor de voltaje para normalizar la señal analógica.
- Interfaz gráfica fácil de configurar e instalar.
- La información registrada en el sensor es rápidamente procesada y transmitida a la interfaz HMI.

Solucion G. Entre las características de esta solución están:

- El sistema se puede implementar fácilmente y a un bajo costo.
- Una de sus desventajas está en las características del sensor como su fragilidad al uso continuo, y necesidad de continuo mantenimiento.
- Solo es necesario realizar un acople de impedancias y un divisor de voltaje para normalizar la señal analógica.
- Interfaz gráfica fácil de configurar e instalar.
- La información registrada en el sensor es rápidamente procesada y transmitida a la interfaz HMI.

Solucion M. Entre las características de esta solución están:

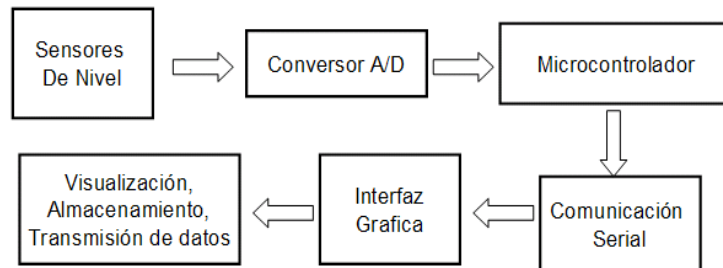
- El sistema se puede implementar fácilmente y a un bajo costo.
- Una de sus desventajas se presenta cuando en algunos casos la salida del sensor debe acondicionarse y normalizarse.
- Solo es necesario realizar un acople de impedancias y un divisor de voltaje para normalizar la señal analógica.
- Interfaz gráfica fácil de configurar e instalar.
- El sensor es uno de los más utilizados para medir líquidos inflamables como la gasolina, el petróleo y el acpm.
- La información registrada en el sensor es rápidamente procesada y transmitida a la interfaz HMI.

3. DISEÑO ARQUITECTURAL

En este capítulo se desarrollara la estructura y la representación grafica de cada uno de los procesos involucrados y sus correspondientes funciones, subfunciones y operaciones (Ver Figura 21).

3.1. DIAGRAMA DE BLOQUES.

Figura 21. Diagrama de bloques del sistema



3.2. DIAGRAMAS DE FLUJO

Diagrama de flujo de los procesos realizados por el microcontrolador (Ver Figura 22).

Figura 22. Diagramas de flujo

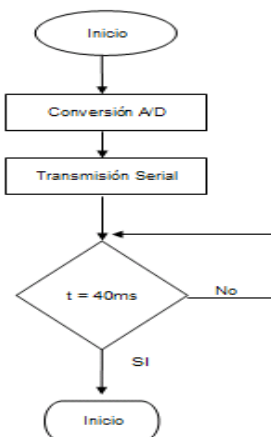


Diagrama de la conversión análoga digital (Ver Figura 23).

Figura 23. Diagrama de la conversión análoga digital

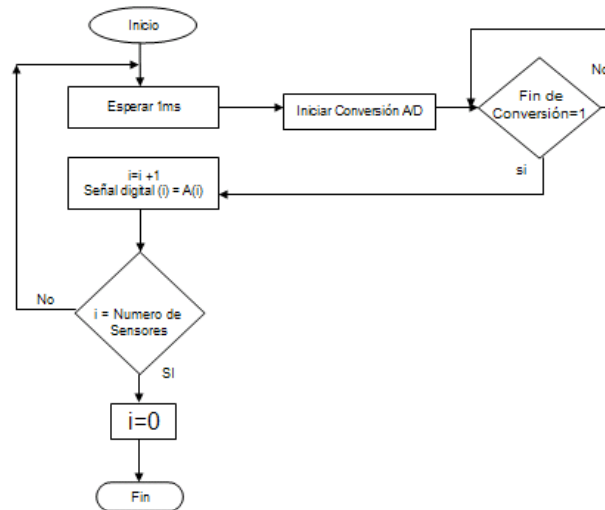


Diagrama transmisión serial (Ver Figura 24).

Figura 24. Diagrama transmisión serial

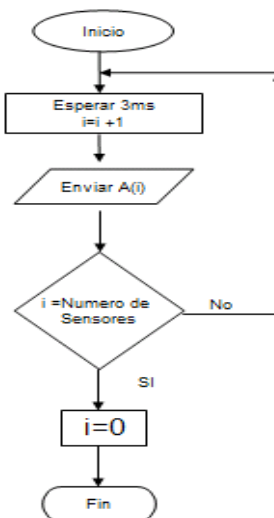


Diagrama de Interfaz Grafica (HMI) (Ver Figura 25).

Figura 25. Diagrama de interfaz grafica (HMI).

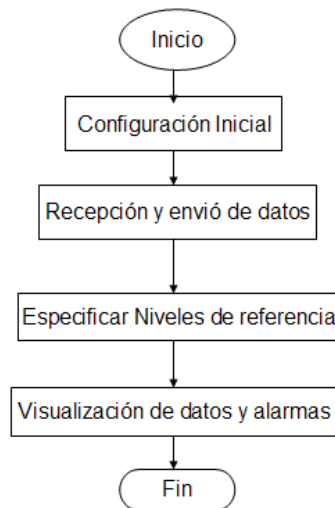


Diagrama de configuración inicial (Ver Figura 26).

Figura 26. Diagrama de configuración inicial.

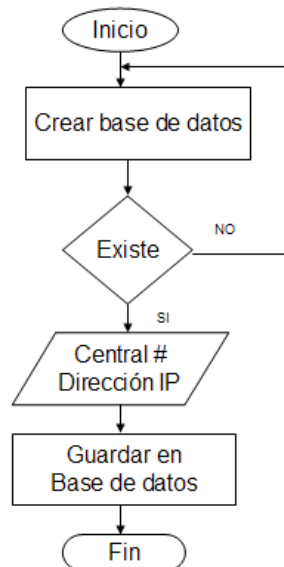


Diagrama de Procesos Recepción y envío de datos (Ver Figura 27).

Figura 27. Diagrama de procesos recepción y envío de datos.

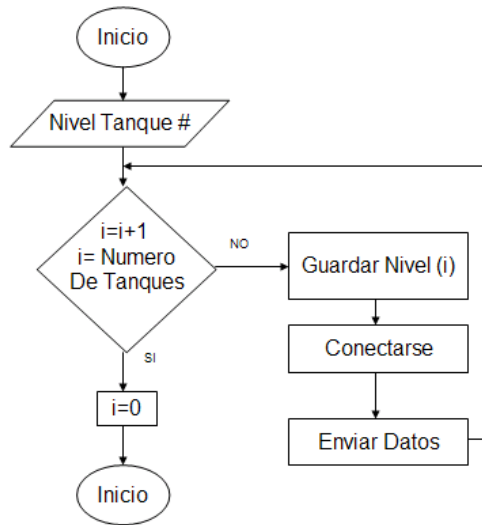


Diagrama Conectarse a otras centrales (Ver Figura 28).

Figura 28. Diagrama conectarse a otras centrales.

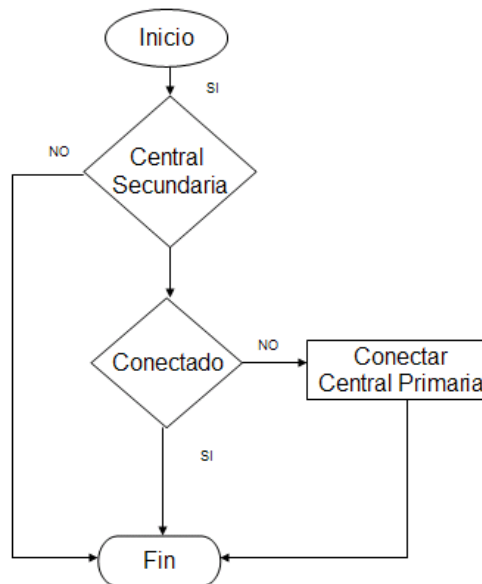


Diagrama Especificar niveles de referencia (Ver Figura 29).

Figura 29. Diagrama especificar niveles de referencia.

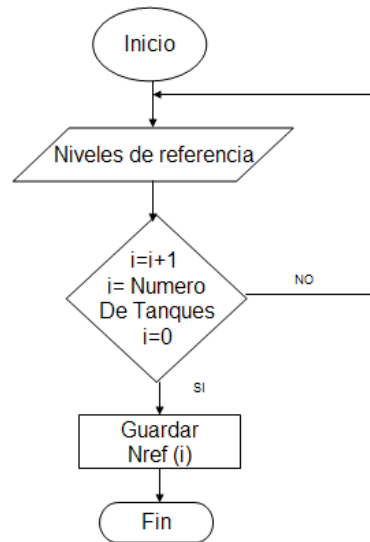
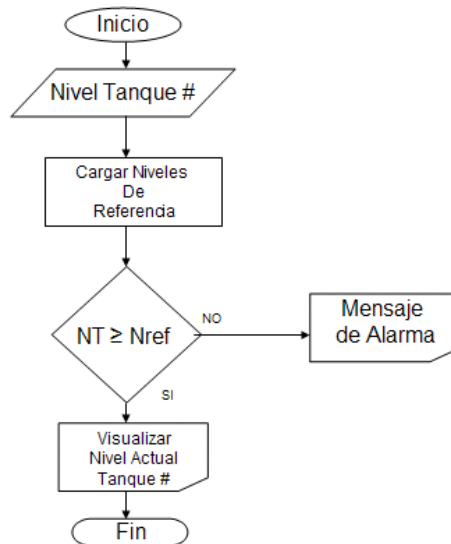


Diagrama Visualización de datos y alarmas (Ver Figura 30).

Figura 30. Diagrama visualización de datos y alarmas.



4. DISEÑO DETALLADO

4.1. DESARROLLO DEL SISTEMA

En este capítulo correspondiente a la tercera etapa del desarrollo del proyecto se diseñaran, programaran y se especificaran las características y funciones del dispositivo en base a la solución escogida en el capítulo de diseño conceptual.

4.1.1. Descripción del sistema. El sistema se diseño y desarrollo con base en los análisis de cada función su correspondiente solución y su representación en diagrama lógicos y arquitecturales.

El sistema se dividió en cinco partes:

- Dispositivo de medición
- Dispositivo comunicación serial
- HMI: Interfaz con el usuario
- Diseño de la alimentación
- Posible ampliación del sistema

El sistema funcionara de la siguiente manera, en cada una de las centrales se registrara por medio de un sensor el nivel de cada uno de los tanques de combustible, la información correspondiente al nivel actual del tanque se digitalizara por medio de un conversor A/D controlado por un microcontrolador. En el microcontrolador se determinara el tiempo de muestreo de las variables recolectadas del conversor A/D para su posterior procesamiento y envío en forma serial al PC.

En el PC por medio de una interfaz grafica el usuario podra visualizar el estado actual de cada uno de los tanques y cualquier falla que se presente en cada uno de ellos. Utilizando la red que conecta las centrales de EMCALI por medio del protocolo TCP/IP, se transmitira información entre los PC de control de cada central secundaria a la central principal de colon con el fin de centralizar el procesamiento y almacenamiento de la información recolectada. Los procesos o eventos ocurridos en cada uno de los tanques se almacenaran en una base de datos permitiendo llevar un control sobre su consumo y comportamiento.

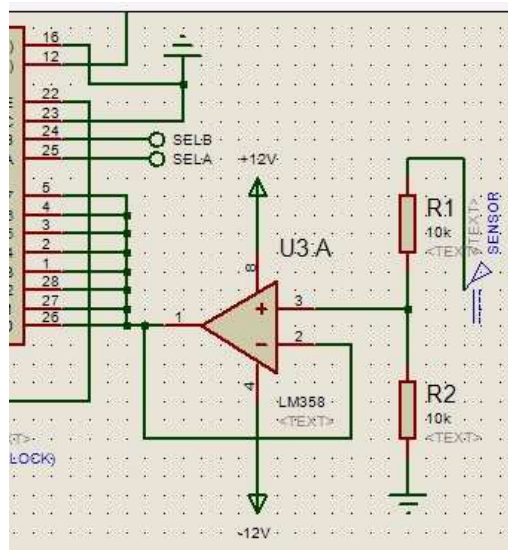
► Dispositivo de medición. La medición el nivel de combustible en los tanques se realizara por medio de un sensor de presión diferencial, el sensor escogido es el **mpx5010**. Ver Anexo 3. Este dispositivo permite medir la presión del tanque comparándola con la presión en una columna de un líquido con una densidad distinta a la densidad del combustible o comparándola con la presión atmosférica fuera del tanque.

Características del Sensor

- Voltaje de alimentación: 5.0 voltios DC
- Rango de medición: 0 a 10kpa
- Span: en el máximo rango de medición, min=4.275, max=4.725 Voltios DC
- Salida: [0 a 5Voltios] o [0 a 10Voltios] DC
- Tiempo de respuesta: 1 Milisegundo.

La señal entregada por el sensor será transmitida hasta el dispositivo de comunicación serial por medio de un cable par trenzado de cobre el cual permite abarcar distancias hasta de 2km sin perdidas en la transmision. El acondicionamiento de la señal del sensor se realizo con un divisor de voltaje y un Lm358 en configuración de buffer o seguidor de voltaje para el acople de impedancias (Ver Figura 31).

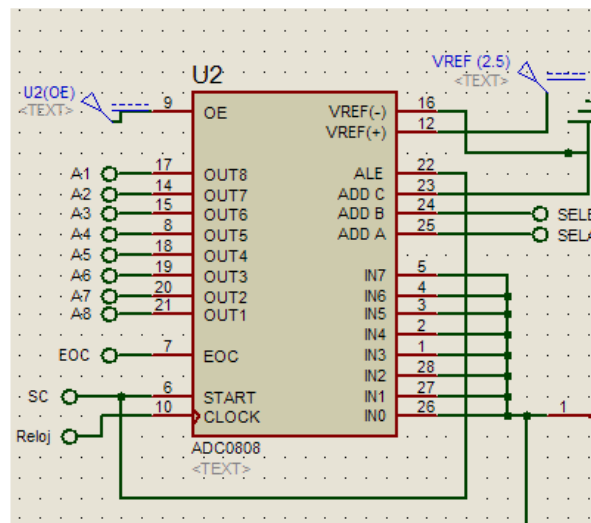
Figura 31. Acondicionamiento de la señal del sensor de presión.



► Dispositivo de comunicación serial. En este modulo se lleva a cabo la conversión de los datos análogos en digitales para su posterior procesamiento y transmisión por comunicación serial a una PC, en el diseño este modulo consta de 8 entradas análogas correspondientes a cada tanque y tiene la posibilidad de ampliarse a un máximo de 16 entradas. Para su implementación se utilizó un microcontrolador Atmel89c52, el conversor análogo-digital ADC0808 y la interfaz de comunicación serial RS-232.

La digitalización se lleva a cabo en el ADC0808 este es controlado por el microcontrolador por medio de la señal de inicio de conversión, y las señales para multiplexar cada una de las 8 salidas correspondientes cada una a un sensor diferente (Ver Figura 32). (Ver anexo E).

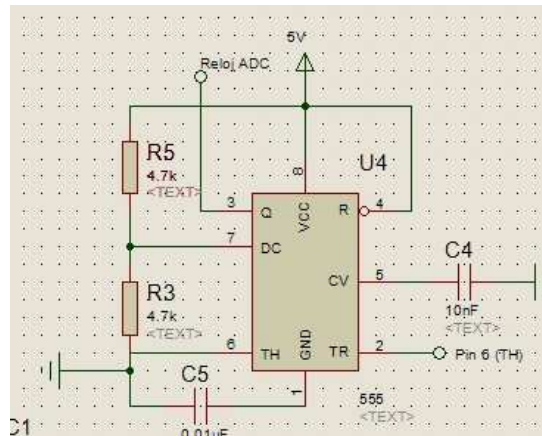
Figura 32. ADC0808.



El tiempo de conversión a una frecuencia mínima de 10Khz es de 90 μ s, y a una frecuencia típica de 640Khz su t_c es de 100 μ s.

Para generar la frecuencia de reloj del ADC se utilizó el integrado LM555 en configuración estable. Características LM555 (Ver Anexo D). (Ver Figura 33).

Figura 33. Configuración LM555.



La frecuencia de salida viene dada por la siguiente formula:

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2 R_B) C}$$

Para una frecuencia de 10Khz y un condensador de 0.01µf el valor de las resistencias es

$$1.44 / f_c = R_A + R_B \rightarrow 1.44 / 0.001 = 14400\Omega$$

$$\text{Si } R_A = 4.7K\Omega \quad R_B = 4.8 K\Omega$$

$$\text{Resolucion} = \frac{V_{in \max}}{2^n - 1}$$

Para este caso la resolución del ADC con respecto a la salida del sensor es:

$$\frac{5V}{2^8 - 1} = 0.019 \frac{mV}{\text{Paso}}$$

Para la elección del microcontrolador este se hizo una comparación del precio. En este caso se comparo el atmel89c52 con el pic16f84, por su funcionalidad y características similares la única desventaja del atmel en comparación al pic es que este trae integrado un conversor análogo digital. (Ver Tabla 19). Comparación precios.

Tabla 19. Comparación precios

Dispositivo	Precio
Atmel89c52	\$ 5.400
Pic 16F84	\$ 9.600
ADC0808	\$ 3.300

El dispositivo de comunicación serial se desarrollo con el microcontrolador atmel 89c52 (Ver Figura 37), y el RS-232. Las velocidades de transmisión de se determinaron configurando el microcontrolador con base en la siguiente tabla (ver Tabla 20).

Tabla 20. Velocidades de transmisión serial atmel89c52.

Velocidad en Baudios	Frecuencia de reloj	SMOD	Temporizador 1		
			C/T	Modo	Valor
Modo 0 max 1MHz	12MHz	x	x	x	x
Modo 2 max: 375 K	12MHz	1	x	x	x
Modo 1,3 : 62,5K	12MHz	1	0	2	FFH
19.200	11,059 MHz	1	0	2	FDH
9.600	11,059 MHz	0	0	2	FDH
4.800	11,059 MHz	0	0	2	FAH
2.400	11,059 MHz	0	0	2	F4H
1.200	11,059 MHz	0	0	2	E8H
137.500	11,059 MHz	0	0	2	1DH
110.000	6 Mhz	0	0	2	72H
110.000	12 Mhz	0	0	1	FE8H

El microcontrolador se configuro para transmitir en modo 2 con una frecuencia de reloj de 11.059 Mhz, de acuerdo a la tabla la velocidad de transmisión será de 9600 baudios. (Ver Anexo B).

Para la transmisión de los datos a la interfaz HMI en el computador se conecta al dispositivo través del puerto serial para lo cual se usa un conector DB-9 conectado al integrado MAX-232 y este conectado al microcontrolador. La función del microcontrolador es la de encargarse de recolectar los datos digitalizados en el ADC, procesarlos y enviarlos en forma serial al integrado max-232.

La función del max-232 es transformar los voltajes usados por el puerto serial del computador que oscilan entre +12 V y -12 V en 0 V y +5V respectivamente, los cuales son comprendidos por el puerto serial del microcontrolador. (Ver Figura 35).

Figura 34. Dispositivo acondicionamiento y comunicación serial.

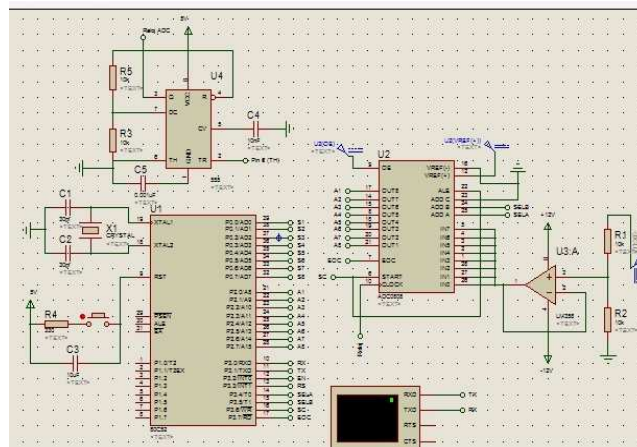


Figura 35. Microcontrolador atmel89c52.

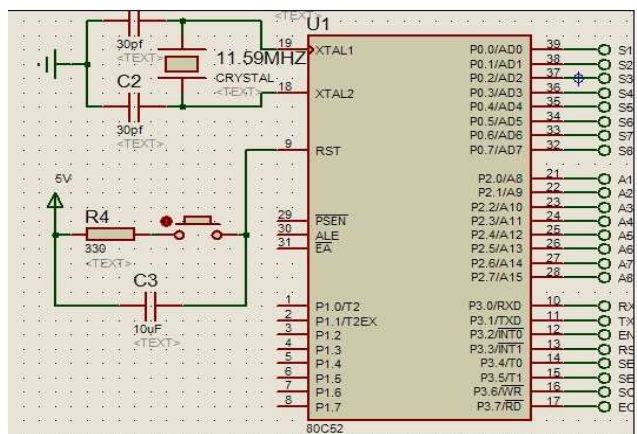
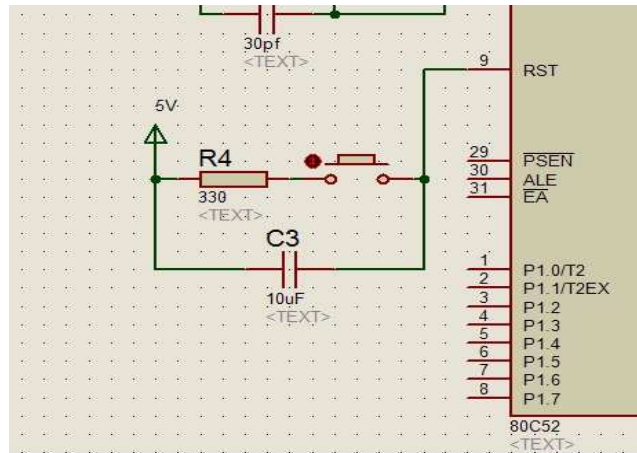


Figura 36. Circuito reset microcontrolador.



► HMI: interfaz con el usuario. Esta interfaz se encarga de permitirle al usuario interactuar con el modulo de comunicación serial y configurar el sistema de monitoreo para registrar en una base de datos los niveles registrados cada cierto tiempo y el historial de eventos de alarma que se puedan presentar.

En el proceso de configuración inicial se crea la base de datos con los campos correspondientes, a cada central, su dirección ip, sus respectivos tanques y eventos de alarma ocurridos durante el día (ver anexo G), (ver Figura 37, 38).

Figura 37. Ventana configuración Inicial.

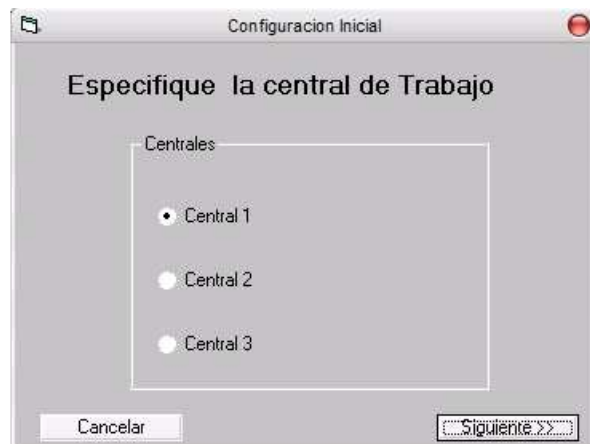
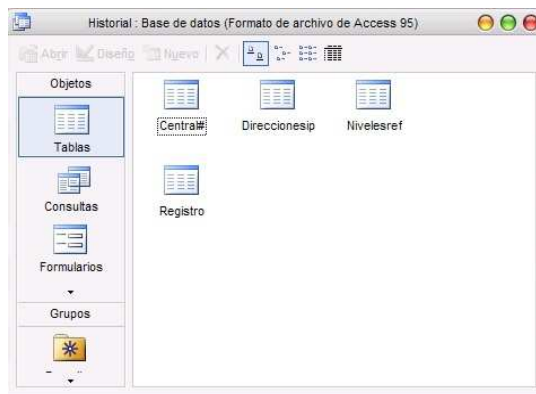


Figura 38. Base de datos.



En el proceso de Direcciones IP se registran y almacenan las direcciones de la central principal o maestra y de las centrales secundarias. (Ver Anexo H), (Ver Figura 40).

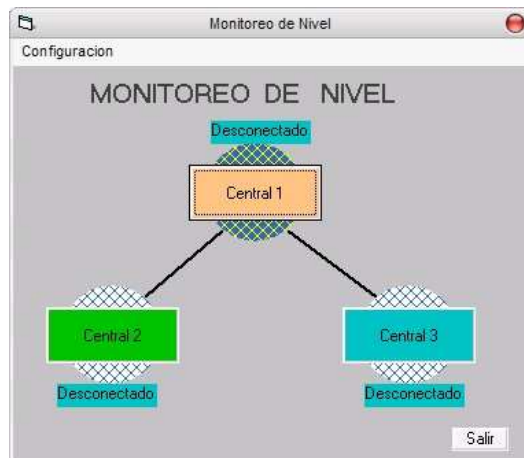
Figura 39. Ventana dirección ip.



En el proceso monitoreo de nivel se lleva a cabo la conexión, entre centrales para el intercambio de información con la central principal. Y se reciben los datos transmitidos al puerto serie desde el microcontrolador a través de la interfaz RS-232. (ver Figura 41). La comunicación entre centrales se realiza por medio de la red interna que posee emcali, en esta red se encuentran ubicados los Pc o puntos de monitoreo los cuales se configuran con una dirección ip estatica para facilitar permittir la comunicación directa entre los puntos de monitoreo.

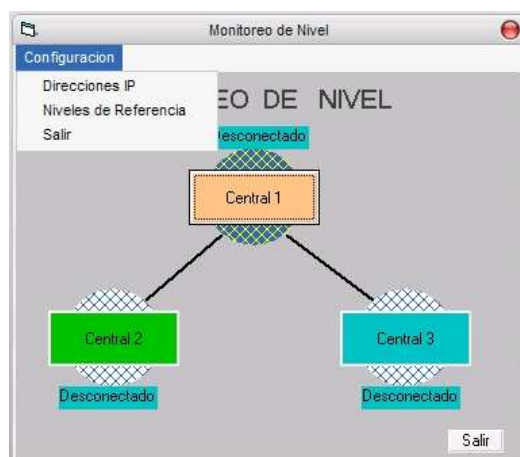
La velocidad de transmisión de esta red varía entre 10Kbps y 10Mbps permitiendo enviar los paquetes de información en forma rápida y que el tráfico de información sea despreciable.

Figura 40. Ventana principal.



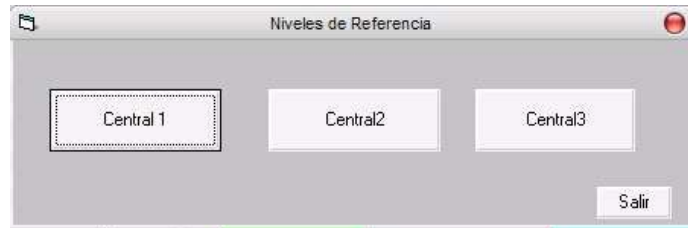
Por medio de la ventana principal se puede acceder al menú configuración donde se encuentra la configuración de las direcciones IP y los niveles de referencia (ver Figura 41), (Ver Anexo J).

Figura 41. Menú de configuración.



A través de la ventana Niveles de referencia se accede a la configuración de los niveles de referencia, para su modificación o consulta (ver Figura 42), (Ver Anexo I)

Figura 42. Ventana niveles de referencia.



A través de las ventanas de nivel de referencia se establece el punto crítico donde se genera la alarma del sistema (ver Figura 43), (Ver Anexo I).

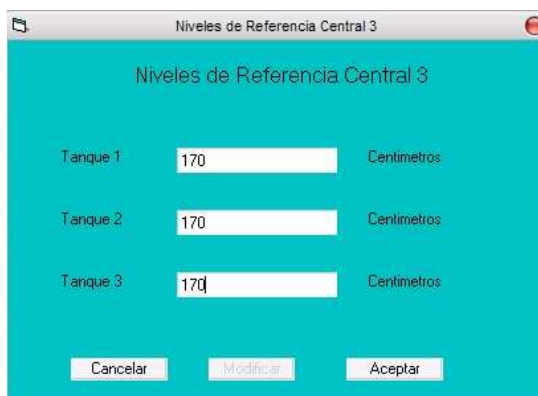
Figura 43. Ventana niveles de referencia central1.

A screenshot of a software window titled "Niveles de Referencia Central1". The background is orange. It displays three rows of input fields for "Tanque 1", "Tanque 2", and "Tanque 3", each with the value "150" and the unit "Centimetros". At the bottom, there are "Cancelar" and "Aceptar" buttons.

Figura 44. Ventana niveles de referencia central2.

A screenshot of a software window titled "Niveles de Referencia Central2". The background is green. It displays three rows of input fields for "Tanque 1", "Tanque 2", and "Tanque 3", each with the value "160" and the unit "Centimetros". At the bottom, there are "Cancelar", "Modificar", and "Aceptar" buttons.

Figura 45. Ventana niveles de referencia central3.



Niveles de Referencia Central 3

Tanque 1 170 Centimetros

Tanque 2 170 Centimetros

Tanque 3 170 Centimetros

Cancelar Modificar Aceptar

A través de la ventanas de monitoreo de cada central se accede a la visualización dinámica de cada uno de los tanques (ver Figuras 44, 45, 46, 47), (Ver Anexo K).

Figura 46. Ventana monitoreo central1.



Monitoreo Central 1

Tanque1 Tanque2 Tanque3

Salir

Figura 47. Ventana monitoreo central1 tanque1.



Figura 48. Ventana monitoreo central2.

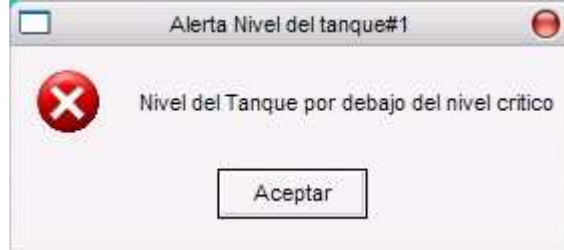


Figura 49. Ventana monitoreo central2 tanque1.



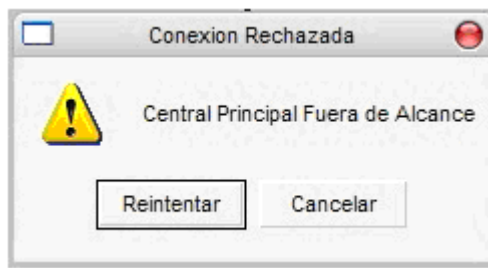
Cuando se presenta una alarma el sistema despliega la siguiente ventana indicando el tanque donde se esta presentando la alerta de forma llamativa (ver Figuras 48, 49, 50), (Ver Anexo K).

Figura 50. Ventana de alerta.



Cuando hay un error en la conexión o la conexión es rechazada el sistema despliega la siguiente ventana preguntando si se quiere seguir tratando de conectar o quiere cancelar la conexión (ver Figura 53), (Ver Anexo K).

Figura 51. Ventana de conexión rechazada.



► Diseño de la alimentación. El dispositivo se ha diseñado para que no solo sea usado en un lugar fijo, sino que pueda ser móvil y portátil, por lo cual su alimentación puede provenir de la línea de corriente alterna 110v o 220V; por medio de un adaptador de voltaje. El módulo de alimentación tiene un conector para la conexión de un adaptador de voltaje estándar al circuito, este adaptador de voltaje será de 9voltios DC y entregara una corriente máxima de 450ma. El dispositivo tendra un interruptor para encender y apagar, y al final estaráa conectado un regulador 7805, el cual regula el voltaje de salida para suministrar cinco voltios al dispositivo. Este regulador posee un sistema de protección para corte del suministro de energía por aumento de temperatura, dando un rango de operación segura, haciendo que el sistema sea inmune a cambios en los voltajes de la red eléctrica o cualquier elemento que se use como alimentación. También limita el paso de corriente en caso de corto circuito, disminuyendo la posibilidad de dañar algún componente del dispositivo.

► **Diseño de impreso del sistema.** Para el diseño impreso se utilizó el software de diseño Eagle 4.16r2, primero se desarrolló el diseño esquemático. (Ver Figura 52). Después automáticamente con ayuda del software se genera el diseño impreso. (Ver Figura 53).

Figura 52. Diseño esquemático del sistema.

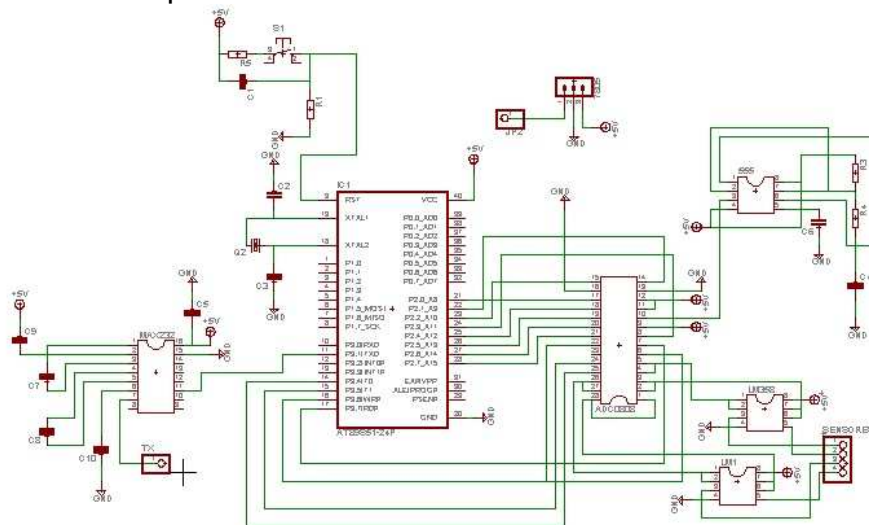
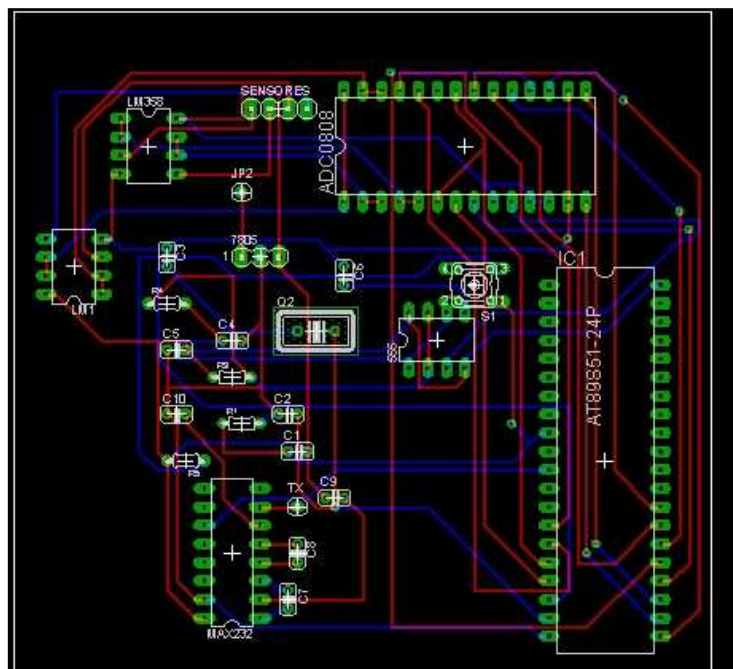


Figura 53. Diseño impreso del sistema.



► Posible ampliación del sistema. La ampliación del sistema se puede implementar fácilmente mediante multiplexores de 8 a 4 en este caso se implemento con el 74LS157. el microcontrolador se encarga de controlar el ADC y los multiplexores para determinar cual de las 16 señales es la que se quiere monitorear. (ver Figura 54). (ver Tabla 21).

Figura 54. Ampliación de la capacidad de tanques registrados

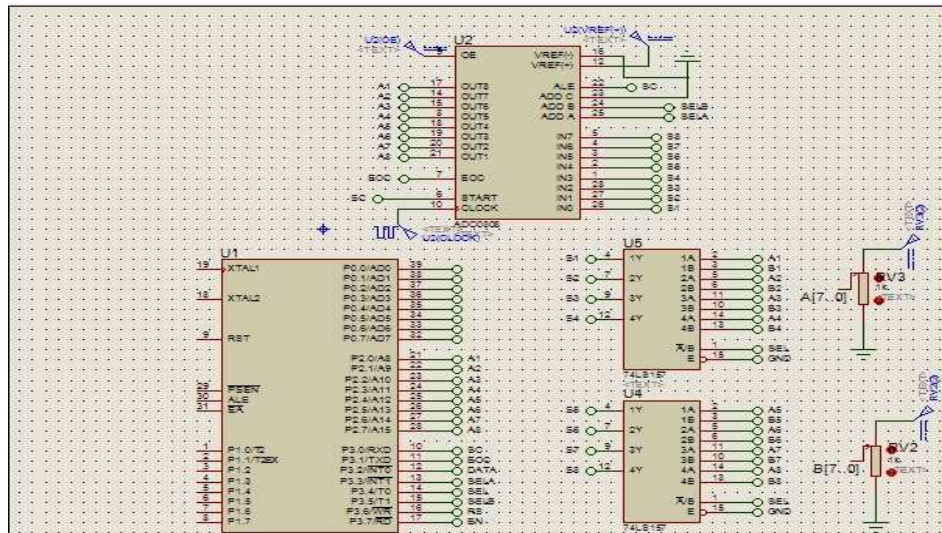


Tabla 21. Selección de la señal de entrada.

Sensor #	Selector multiplexor	Selector C ADC	Selector B ADC	Selector A ADC
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	0	0
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

► Diseño del prototipo. El prototipo físico tendrá las siguientes medidas: Alto 3cm, Ancho 8cm, Largo 10cm, cumpliendo con el requerimiento de transportabilidad y ocupar el mínimo espacio de trabajo. El prototipo tiene un botón de encendido, un botón de reset, un led indicador de funcionamiento, puerto para conexión serial, entrada para el adaptador de voltaje y 8 entradas correspondientes a las señales de cada sensor de nivel. (ver Figura 54).

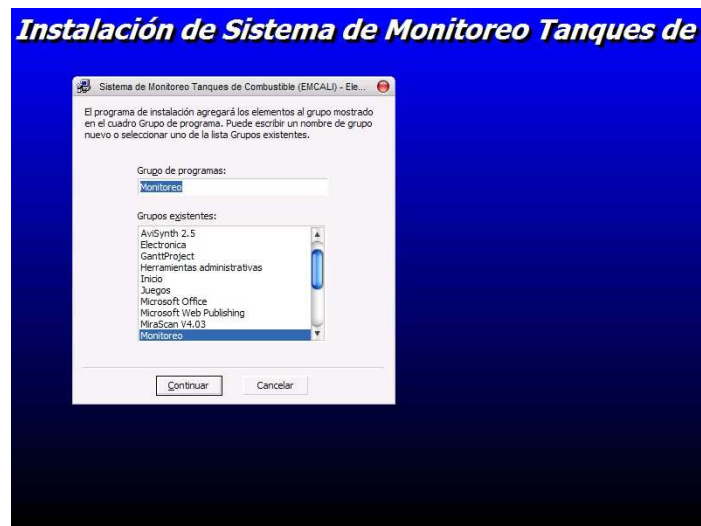
Figura 55. Diseño del prototipo del dispositivo de comunicación.



En esta prueba solo la Central principal recibía datos reales, por que solo se cuenta con una planta de nivel en el laboratorio, para las otras centrales se generaron aleatoriamente valores entre 1 y 180 (Ver anexo K). Con el resultado de esta pruebas y el constante uso de la interfaz grafica se realizo un manual de usuario de la configuración de esta interfaz y su funcionamiento (Ver anexo A).

Para cumplir con los requerimientos de portabilidad, facilidad de uso e instalación se genero un programa de instalación que permite que funciones desde cualquier PC con plataforma Windows, (ver Figura 57).

Figura 57. Instalación de Sistema de Monitoreo.



5.1 COSTO DEL DESARROLLO DEL PROTOTIPO

Los gastos estimados para la realización de este proyecto son sugeridos a continuación (ver Tabla 22).

Tabla 22. Gastos estimados para la realización del proyecto.

Ítem	Costo
1. Director de proyecto.	\$ 800.000.00
2. Computadoras.	\$ 3.000.000.00
3. Impresión (informes, bibliografía.)	\$ 70.000.00
4. Papelería (Fotocopias, hojas, lápiz, borradores, etc.)	\$ 65.000.00
5. Internet.	\$ 250.000.00
6. Elementos de laboratorio para pruebas	\$ 30.000.00
7. Gastos Varios (Teléfono, transporte, alimentación)	\$ 850.000.00
8. Costos de Ingeniería (\$15.000/ hora de trabajo)	\$ 13.500.000.00
Total	\$ 18.565.000.00

Costo de los elementos del Prototipo (ver Tabla 23).

Tabla 23. Costo de los elementos del Prototipo.

Elemento	Precio en pesos
Atmel89c52	\$ 5.400
Adc0808	\$ 3.300
Rs-232	\$ 2.300
Metro Cable serial	\$ 4.000
555	\$ 500
Resistencias y condensadores	\$ 700
7805	\$ 2.000
Adaptador 9v DC, 500ma	\$ 15.000
LM358	\$ 1.000
TOTAL	\$ 34.200

5.2 CONSUMO DE POTENCIA DEL PROTOTIPO.

Tabla 24. Consumo de Potencia del Dispositivo.

Elemento	Corriente máxima	Potencia consumida
Atmel89c52	25ma	125 watios
Adc0808	3ma	15 watios
Rs-232	15ma	75 watios
555	25ma	125 watios
TOTAL	68ma	340 watios

6. CONCLUSIONES

- Se desarrollo el diseño de un sistema que permite almacenar y consultar el historial de los tanques de combustible de las diferentes centrales de EMCALI.
- Se desarrollo el diseño de un sistema eficiente de bajo costo, altamente configurable, fácil de manejar y entender.
- El diseño del sistema cumple con los requerimientos planteados en el capitulo2, y con las características de un sistema SCADA centralizado.
- El costo de diseño es bastante económico lo cual hace que el diseño se pueda implementar fácilmente en cualquier central.
- Como se puede verificar, la forma y las dimensiones del prototipo son muy reducidas, por lo tanto no presentará inconvenientes técnicos, de espacio o de transporte.
- Como resultado de las pruebas realizadas tanto en el software como en el hardware, la interfaz grafica y los dispositivos de comunicación ejecutaron correctamente sus funciones, esto demuestra que son herramientas útiles, confiables y seguras.

7. RECOMENDACIONES

- Ubicar el centro de trabajo y monitoreo cerca de los tanques para evitar pérdidas en la transmisión de datos.
- Ubicar el centro de monitoreo de forma centralizada de tal forma que encuentre a igual distancia de los tanques de combustible.
- Todos los usuarios de este sistema de monitoreo deben familiarizarse con el entorno grafico y el dispositivo de transmisión antes de empezar a realizar el monitoreo de cualquier tanque de combustible.

BIBLIOGRAFIA

BELTRÁN PROVOSTE, Cristhian. Medición de nivel [en línea]. Barcelona: infoPLC.net, 2008. Modificada por última vez 13 septiembre 2007. [Consultado 03 Marzo de 2007]. Disponible Internet: http://www.infoplcn.net/Documentacion/Docu_Instrumentacion/infoPLC_net_MEDICION_DE%20NIVEL.pdf

COMER, Douglas E. Redes globales de informacion con internet y tcp/ip. 3 ed. Naucalpan de Juarez. Estado de Mexico: Prentice Hall, 1996. 621 p.

Familia de protocolos de Internet [en línea]. Florida: Wikimedia Foundation, 2006. Modificada por última vez 7 febrero 2008. [Consultado 11 Noviembre de 2007]. Disponible en Internet: http://es.wikipedia.org/wiki/Familia_de_protocolos_de_Internet

HERMAN, Mark Steven. La esencia de visual basic. 4 ed. Naucalpan de Juarez. Estado de Mexico: Prentice Hall, 1996. 457 p.

PALLAS, Ramón, Sensores y acondicionadores de señal. 3 ed. Mexico: Alfaomega, 2001. 480 p.

STALLINGS, William. Comunicaciones y redes de computadores. 6 ed. Madrid, España: Prentice Hall, 1997. 729 p.

Transmission Control Protocol [en línea]. Florida: Wikimedia Foundation, 2008. Modificada por última vez 30 enero 2008. [Consultado 11 Noviembre de 2007]. Disponible en Internet: <http://es.wikipedia.org/wiki/TCP>

VIERA CHILE, Iremis. Sistemas de Adquisición de Datos [en línea]. Cuba: Universidad "Hermanos Saiz Montes de Oca". Modificada por última vez 29 diciembre 2004. [Consultado 05 Noviembre de 2007]. Disponible en Internet: <http://www.monografias.com/trabajos17/sistemas-adquisicion-dato/sistemas-adquisicion-dato.shtml>

ANEXOS

Anexo A. Manual de usuario

1. Configuración inicial

- En esta ventana aparecerán las posibles ubicaciones de la central primaria y las centrales secundarias. Se debe marcar correctamente para evitar errores en la conexión. Si se quiere salir sin realizar ninguna modificación pulse el botón **Cancelar**.
- Si considera que a marcado correctamente las casilla pulse el boton **Siguiente**, [Figura 1. Configuración Inicial].

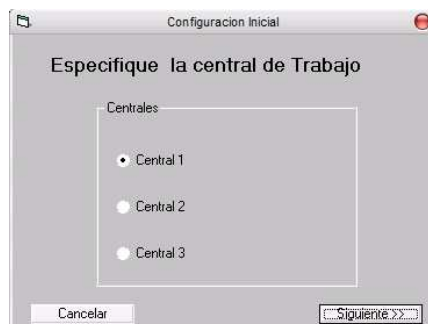


Figura (1). Configuración Inicial.

2. Direcciones ip

- Si considera que se equivoco en la configuración anterior puede pulsar el botón **Atrás**. o puede seguir hasta la ventana principal y reconfigurarlo por medio del **menú configuración**.
- En esta ventana aparecen tres casillas que se deben llenar correctamente para evitar fallas constantes en la conexión entre una central y otra. Recuerde que la dirección ip se compone de 4 números que tiene un valor máximo de 255 separados por puntos
- Después de terminar de llenar las casilla correctamente por favor pulsar el botón **Siguiente** [Figura 2].



Figura (2). Dirección IP

3. Monitoreo de Centrales

- Esta ventana es la parte principal de toda la interfaz grafica, desde esta ventana podrá acceder a la información de la central principal pulsando el botón **Central 1** [Figura 3].
- Podrá acceder a información de la central # 2 principal pulsando el botón **Central 2** [Figura 3].
- Podrá acceder a información la central # 3 principal pulsando el botón **Central 3** [Figura 3].
- Si desea modificar alguno de los parámetros anteriormente del sistema anteriormente especificados pulse en el menú configuración en cualquiera de las opciones que aparecen [Figura 3].
- Si desea salirse del sistema pulse el botón **Salir** en la ventana principal o en el menú configuración [Figura 3].

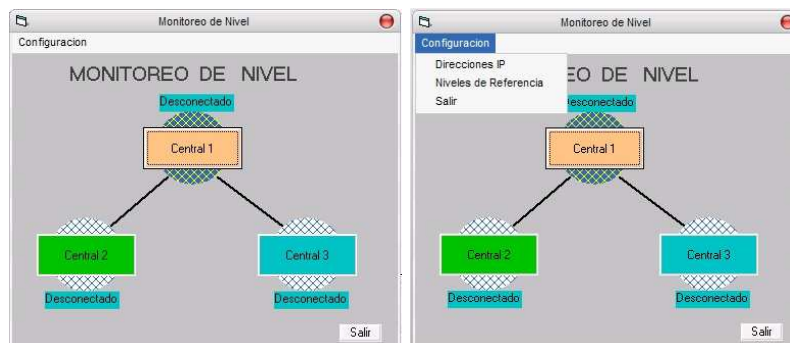


Figura (3). Monitoreo de Centrales

4. Monitoreo de los tanques

- Si desea visualizar alguno de los tanques pulse en cualquiera de los botones con el nombre **Tanque #**. [Figura 4].
- Si desea cerrar de esta ventana pulse el botón **Salir**. [Figura 4].



Figura (4). Monitoreo de los tanques

5. Visualización del nivel actual de los tanques

- Si desea cerrar esta ventana pulse **Salir** [5].



Figura (5). Visualización del nivel actual de los tanques

6. Control de alarmas y errores de conexión

- Si se le presenta una ventana de alerta, revise y atienda manualmente el problema. [Figura 6].

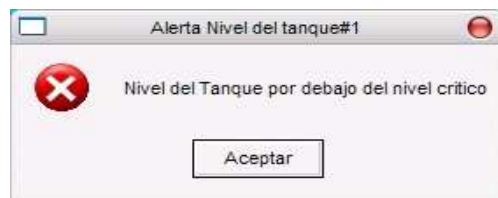


Figura (6). Control de alarmas y errores de conexión

7. Reiniciar

- Si se le presenta un error en la conexión es posible que la central principal no se encuentre en línea o haya rechazado la conexión si quiere reintentarlo pulse el boton **Reintentar** si desea cancelar la conexión pulse el boton **Cancelar**. [Figura 7].

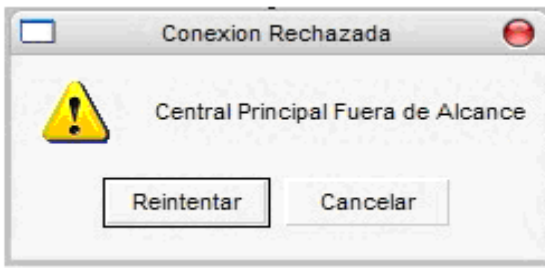


Figura (7). Reintentar

Nota: se le permite al usuario abrir múltiples ventanas correspondientes a cada una de las centrales

Anexo B. Características Microcontrolador atmel89c52.

Features

- Compatible with 80C51TM Products
- 8 Kbytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-Bit Timer/Counters
- Eight Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

Description

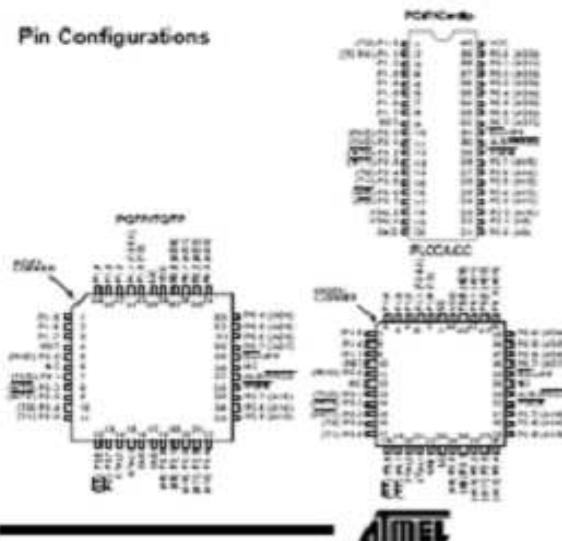
The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8 Kbytes of Flash programmable and erasable read-only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry standard 80C51 and 80C52 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer which provides a highly flexible and cost-effective solution to many embedded control applications.

The AT89C52 provides the following standard features: 8 Kbytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89C52 is

(continued)

8-Bit Microcontroller with 8 Kbytes Flash

Pin Configurations





Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground.....	-1.0 V to +7.0 V
Maximum Operating Voltage.....	6.6 V
DC Output Current.....	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

The values shown in this table are valid for $T_A = -40^{\circ}\text{C}$ to 85°C and $V_{CC} = 5.0\text{ V} \pm 20\%$, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low Voltage (EA)		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽¹⁾ (Ports 1, 2, 3)	$I_{OL} = 1.8\text{ mA}$		0.45	V
V_{OL1}	Output Low Voltage ⁽¹⁾ (Port 0, ALE, PSEN)	$I_{OL} = 3.2\text{ mA}$		0.45	V
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE, PSEN)	$I_{OH} = -60\text{ }\mu\text{A}$, $V_{CC} = 5\text{ V} \pm 10\%$	2.4		V
		$I_{OH} = -25\text{ }\mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10\text{ }\mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800\text{ }\mu\text{A}$, $V_{CC} = 5\text{ V} \pm 10\%$	2.4		V
		$I_{OH} = -300\text{ }\mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80\text{ }\mu\text{A}$	$0.9 V_{CC}$		V
I_L	Logical 0 Input Current (Ports 1, 2, 3)	$V_{IH} = 0.45\text{ V}$	-60		μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)	$V_{IH} = 2\text{ V}$	-650		μA
I_L	Input Leakage Current (Port 0, EA)	$0.45 < V_{IH} < V_{CC}$		± 10	μA
RRST	Reset Pulldown Resistor		60	300	k Ω
C_{in}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^{\circ}\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power Down Mode ⁽²⁾	$V_{CC} = 5\text{ V}$ $V_{CC} = 3\text{ V}$		100 40	μA

Anexo C. Características Sensor de presión diferencial.

SEMICONDUCTOR TECHNICAL DATA

SENSOR, STATE MONITORING
by MPX5010/D

Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated

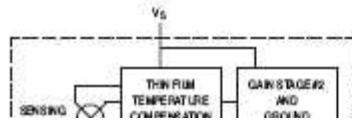
The MPX5010/MPXV5010G series piezoresistive transducers are state-of-the-art monolithic silicon pressure sensors designed for a wide range of applications, but particularly those employing a microcontroller or microprocessor with A/D inputs. This transducer combines advanced micromachining techniques, thin-film metallization, and bipolar processing to provide an accurate, high level analog output signal that is proportional to the applied pressure.

Features

- 5.0% Maximum Error over 0° to 85°C
- Ideally Suited for Microprocessor or Microcontroller-Based Systems
- Durable Epoxy Unibody and Thermoplastic (PPS) Surface Mount Package
- Temperature Compensated over -40° to +125°C
- Patented Silicon Shear Stress Strain Gauge
- Available in Differential and Gauge Configurations
- Available in Surface Mount (SMT) or Through-hole (DIP) Configurations

Application Examples

- Hospital Beds
- HVAC
- Respiratory Systems
- Process Control



SMALL OUTLINE PACKAGE



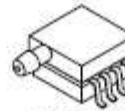
MPXV501008U
CASE 482



MPXV5010GC8U
CASE 482A



MPXV5010GC7U
CASE 482C



MPXV5010GP
CASE 1388



MPX5010 MPXV5010G SERIES

Motorola Preferred Device

INTEGRATED
PRESSURE SENSOR
0 to 10 kPa (0 to 1.45 psi)
0.2 to 4.7 V Output

UNIBODY PACKAGE



MPX5010D
CASE 887



MPX5010DP
CASE 887C



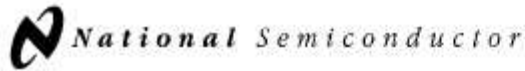
Freescale Semiconductor, Inc.

OPERATING CHARACTERISTICS ($V_S = 5.0$ Vdc, $T_A = 25^\circ\text{C}$ unless otherwise noted, $P1 > P2$. Decoupling circuit shown in Figure 3 required to meet specification.)

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range ⁽¹⁾	P_{OP}	0	—	10	kPa
Supply Voltage ⁽²⁾	V_S	4.75	5.0	5.25	Vdc
Supply Current	I_o	—	5.0	10	mAdc
Minimum Pressure Offset ⁽³⁾ @ $V_S = 5.0$ Volts	V_{off}	0	0.2	0.425	Vdc
Full Scale Output ⁽⁴⁾ @ $V_S = 5.0$ Volts	V_{FSO}	4.475	4.7	4.925	Vdc
Full Scale Span ⁽⁵⁾ @ $V_S = 5.0$ Volts	V_{FSS}	4.275	4.5	4.725	Vdc
Accuracy ⁽⁶⁾	—	—	—	± 5.0	% V_{FSS}
Sensitivity	V/P	—	450	—	mV/kPa
Response Time ⁽⁷⁾	t_R	—	1.0	—	ms
Output Source Current at Full Scale Output	I_{O+}	—	0.1	—	mAdc
Warm-Up Time ⁽⁸⁾	—	—	20	—	ms
Offset Stability ⁽⁹⁾	—	—	± 0.5	—	% V_{FSS}

NOTES:

Anexo D. Características LM555.



May 1997

LM555/LM555C Timer

LM555/LM555C Timer

General Description

The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For astable operation as an oscillator, the free running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output circuit can source or sink up to 200 mA or drive TTL circuits.

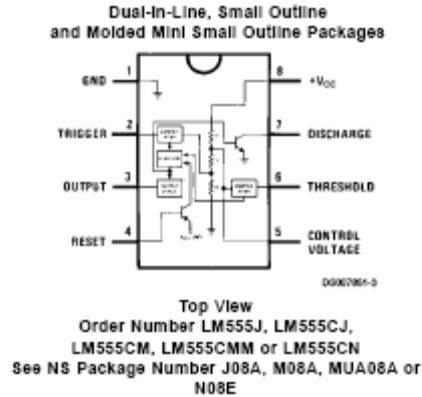
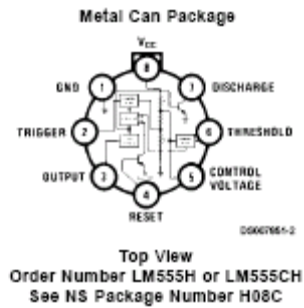
Features

- Direct replacement for SE555/NE555
- Timing from microseconds through hours
- Operates in both astable and monostable modes
- Adjustable duty cycle
- Output can source or sink 200 mA
- Output and supply TTL compatible
- Temperature stability better than 0.005% per °C
- Normally on and normally off output
- Available in 8 pin MSOP package

Applications

- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation
- Pulse position modulation
- Linear ramp generator

Connection Diagrams



Applications Information (Continued)

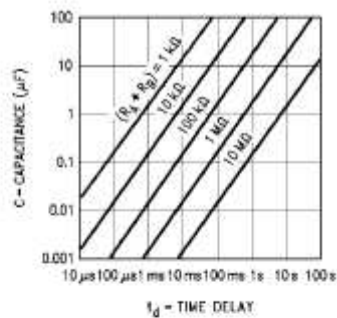


FIGURE 3. Time Delay

ASTABLE OPERATION

If the circuit is connected as shown in Figure 4 (pins 2 and 6 connected) it will trigger itself and free run as a multivibrator. The external capacitor charges through $R_A + R_B$ and discharges through R_B . Thus the duty cycle may be precisely set by the ratio of these two resistors.

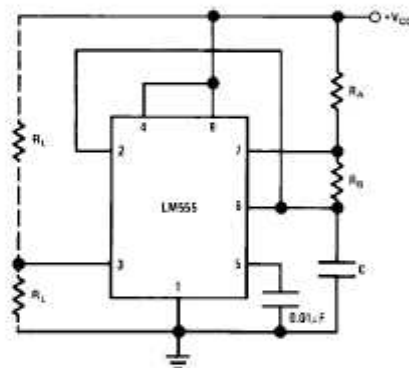
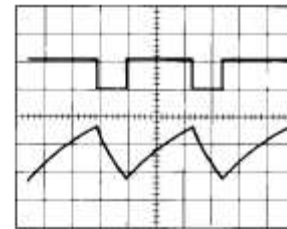


FIGURE 4. Astable

In this mode of operation, the capacitor charges and discharges between $1/3 V_{CC}$ and $2/3 V_{CC}$. As in the triggered mode, the charge and discharge times, and therefore the frequency are independent of the supply voltage.

Figure 5 shows the waveforms generated in this mode of operation.



$V_{CC} = 5V$
 TIME = 20 μs /DIV.
 $R_A = 3.9 k\Omega$
 $R_B = 3 k\Omega$
 $C = 0.01 \mu F$

FIGURE 5. Astable Waveforms

The charge time (output high) is given by:

$$t_1 = 0.693 (R_A + R_B) C$$

And the discharge time (output low) by:

$$t_2 = 0.693 (R_B) C$$

Thus the total period is:

$$T = t_1 + t_2 = 0.693 (R_A + 2R_B) C$$

The frequency of oscillation is:

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B) C}$$

Figure 6 may be used for quick determination of these RC values.

The duty cycle is:

$$D = \frac{R_B}{R_A + 2R_B}$$

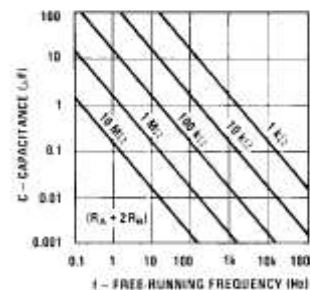














FIGURE 6. Free Running Frequency

FREQUENCY DIVIDER

The monostable circuit of Figure 1 can be used as a frequency divider by adjusting the length of the timing cycle. Figure 7 shows the waveforms generated in a divide by three circuit.

Anexo E. Hoja de seguridad ACPM

HOJA DE SEGURIDAD		
	<p align="center">ACPM</p> <p>Combustible Diesel; Diesel fueloil; aceite combustible N° 2; destilado medio; aceite de calefacción doméstica; aceite combustible para motores Diesel.</p> <p>Mezcla compleja de hidrocarburos entre C₆ y C₂₀, principalmente alifáticos y en menor proporción olefinicos, náticos y aromáticos. Puede contener aditivos (detergentes, etc).</p> <p>Líquido aceitoso incoloro o amarillo pálido, claro y brillante con olor a petróleo. Puede estar coloreado para identificación.</p> <p>CAS[68476-30-2, 68476-34-6, 68334-30-5, 68474-34-6]</p>	 
<p align="center">UN 1202</p>		
<p>RIESGOS Y PRECAUCIONES:</p> <p>Líquido y vapor combustibles. Peligro moderado de incendio.</p> <p>Inhalación de vapor a altas concentraciones puede causar mareos y adormecimiento. El líquido puede producir irritación de la piel y los ojos.</p> <p>Peligro de aspiración si es ingerido. Posibles efectos retardados. Algunos de sus componentes pueden causar cáncer según ensayos con animales.</p> <p>NFPA: Salud 1; Inflamabilidad 2; Reactividad 0</p>		
<p>PROPIEDADES FÍSICOQUÍMICAS IMPORTANTES:</p> <p>Intervalo de ebullición: Oscila entre 215 y 380 °C dependiendo del fabricante.</p> <p>Gravedad específica: 0.83-0.88 a 15°C (agua=1) Menos pesado que el agua.</p> <p>Velocidad de evaporación: Baja, < 0.1 (Acetato de butilo=1).</p> <p>Temperatura de inflamación: 54 °C.</p> <p>Límites de explosividad: Inferior: 1,3 %. Superior: 6%</p> <p>Solubilidad: Solubilidad en agua despreciable (<0.1%). Soluble en éter, cloroformo y solventes derivados del petróleo.</p> <p>Fórmula: Mezcla compleja de hidrocarburos.</p> <p>Densidad del vapor: 3 a 4 (aire=1). Más pesado que el aire</p> <p>Valor de pH: Neutro. Puede contener aditivos que afecten este valor.</p> <p>Temperatura de autoignición: 230 °C.</p> <p>Umbral de olor: 0,1 ppm.</p>		
<p>PRIMEROS AUXILIOS:</p> <p>Asegúrese de que las personas encargadas de aplicar los primeros auxilios no corren riesgo (retire la fuente de contaminación y utilice equipo de protección adecuado).</p> <p>Inhalación: Retire la víctima de la fuente de exposición y llévela al aire fresco. Si no respira, despeje las vías respiratorias y provea resucitación cardiopulmonar si está capacitado para hacerlo. Evite el contacto directo boca a boca. Si la víctima respira con dificultad, personal entrenado debe administrar oxígeno con monitoreo posterior del afectado en forma continua. Obtenga atención médica de inmediato.</p> <p>Contacto con la piel: Retire rápidamente el exceso del producto. Lave por completo el área contaminada con abundante agua y jabón durante por lo menos 15 minutos. Debajo de la corriente de agua retire la ropa, zapatos y artículos de cuero que estén contaminados. No intente neutralizar con agentes químicos. Obtenga atención médica si persiste la irritación.</p> <p>Ingestión: Si la víctima está consciente, no convulsiona y puede ingerir líquido dele a beber dos vasos de agua. No induzca al vómito. Si ocurre vómito espontáneo mantenga la víctima inclinada para reducir el riesgo de aspiración, repita la administración de agua y observe si se presenta dificultad para respirar. Obtenga ayuda médica de inmediato.</p> <p>Contacto con los ojos: Lave de inmediato con abundante agua a baja presión y tibia, preferiblemente, durante por lo menos 15 minutos. Durante el lavado separe los párpados para facilitar la penetración del agua. No intente neutralizar con agentes químicos o gotas sin la orden de un médico. Obtenga atención médica rápidamente.</p>		
<p>INCENDIO:</p> <p>Consideraciones especiales: Líquido combustible. Puede formar mezclas explosivas a temperaturas iguales o superiores a su punto de inflamación. El líquido puede acumular cargas estáticas por transvase o agitación. Los vapores pueden desplazarse a nivel del suelo hasta una fuente de ignición y devolverse ardiendo hasta su lugar de origen. El líquido puede flotar sobre el agua hasta una fuente de ignición y regresar en llamas. El vertimiento del producto a desagües puede causar peligro de fuego o explosión. Produce gases tóxicos por combustión.</p> <p>Procedimiento: Evacúe el área del incendio en 25 a 50 metros en todas direcciones. Si hay un contenedor o carro tanque involucrado, evacúe en 800 metros. Si hay fuga del producto, deténgala antes de intentar apagar el fuego, si puede hacerlo en forma segura. Enfríe los contenedores con agua en forma de rocío, y retírelos del fuego si puede hacerlo sin peligro. No introduzca agua a los contenedores. El agua puede ser inefectiva para extinguir el fuego, dado el bajo punto de inflamación del producto. Aproxímese al fuego en la misma dirección del viento. Para incendios masivos utilice boquillas con soportes. Aléjese de los extremos de los contenedores.</p> <p>Utilice equipo de respiración autocontenido. La ropa normal de bomberos proporciona protección limitada para este producto y sólo se recomienda para operaciones rápidas de entrada-salida en casos especiales.</p> <p>Medios de extinción apropiados: Fuegos pequeños: dióxido de carbono, polvo químico seco, espuma regular.</p> <p>Fuegos grandes: espuma, agua en forma de rocío o niebla. No use agua en forma de chorro.</p>		
<p>VERTIDO ACCIDENTAL:</p> <p>Ubíquese en la dirección desde donde sopla el viento. Evite zonas bajas. Elimine toda fuente de ignición como llamas o chispas.</p> <p>Detenga o controle la fuga, si puede hacerlo sin peligro. Ventile la zona del derrame. No use palas metálicas.</p> <p>Derrames Pequeños: Evacúe y aisle de 25 a 50 metros. Contenga el derrame con diques de poliuretano o calcoetines especiales para aceites y absorba con absorbentes inertes como calcoetines, almohadillas o tapetes para hidrocarburos, chemizorb o vermiculita. Introduzca en contenedores cerrados y marcados. Lave el área con agua y jabón.</p> <p>Derrames Grandes: Evacúe y aisle el área 300 metros en todas direcciones. Utilice agua en forma de rocío para enfriar y dispersar los vapores y proteger al personal. Evite que el material derramado caiga en fuentes de agua, desagües o espacios confinados. Para ello disponga de diques prefabricados. Contacte organismos de socorro.</p> <p>Vertimiento en agua: Utilice absorbentes especiales tipo espagueti para retirar el hidrocarburo de la superficie. Consulte con las autoridades ambientales sobre la posibilidad de utilizar agentes dispersantes o de hundimiento.</p>		

ESTABILIDAD Y REACTIVIDAD: Estable en condiciones normales. Incompatible con oxidantes fuertes (como ácido nítrico, hipoclorito de sodio). No corrosivo a los metales. Productos de descomposición: monóxido de carbono, dióxido de carbono, hidrocarburos reactivos. Condiciones a evitar: Fuentes de ignición como chispas, llamas abiertas y calor intenso. Acumulación de cargas electrostáticas.		
PARAMETROS DE CONTROL POR EXPOSICIÓN: TWA: 5 mg/m ³ (para neblinas de aceite mineral medianamente refinadas) (ACGIH). IDLH = 10 mg/m ³		
MANIPULACIÓN Y ALMACENAMIENTO:		
 >8hr: Caucho de nitrilo, vitón.	 >8hr: Caucho de nitrilo	 Ducha
 Si la protección respiratoria no incluye pieza facial completa, gafas de seguridad para químicos a prueba de salpicaduras, o monogafas.	 Lavacjos	
 De 2 a 10 mg/m ³ : respirador purificador de aire con media máscara y cartucho para vapores orgánicos.		
 10 mg/m ³ : respirador con línea de aire (SAR) operado en modo de flujo continuo, o demanda de presión.		
 Mayores que 10 mg/m ³ o desconocidas: Equipo de respiración autocontenido.		
 Para evitar contacto prolongado o repetido: >8hr: Vitón.		
Riesgo leve o moderado de salpicaduras, traje en Tyvek; ante riesgo alto, Tyvek-Saranex laminado, CPF1, 2, 3 o 4, Responder, Reflector. Evite toda fuente de ignición (chispas, llamas, calor, cigarrillos encendidos). Conecte a tierra contenedores y tuberías. Use sistemas a prueba de chispas y de explosión. Evite generar vapores o neblinas. Nunca realice operaciones de sifón con la boca. Lávese completamente las manos después de su manipulación. Evite contacto con los ojos, la piel y la ropa.		
Almacene bien cerrado en lugar bien ventilado, alejado de materiales incompatibles y calor. A temperatura ambiente (entre 15 y 25 °C). El almacenamiento de grandes cantidades se recomienda en exterior, o en tanques-depositos enterrados. El almacenamiento interno debe hacerse en recinto estándar para líquidos inflamables. Señalice adecuadamente las áreas de almacenamiento y los contenedores.		
TRANSPORTE		
La etiqueta del vehículo debe medir por lo menos 25 cm de lado. Los colores deben ser vivos y la letra contrastante y fácil de leer. Etiquete adecuadamente los contenedores o carotantes y manténgalos cerrados. No lo transporte junto con productos explosivos de las clases 1.1, 1.2, 1.3, 1.5, gases venenosos (2.3), venenosos (6.1). Puede transportarse junto con las clases 5.1 (oxidantes) 1.4 (explosivo) sólo si están separados de tal manera que no se mezclen en caso de derrame. Apague el motor cuando cargue y descargue (si no requiere poner a funcionar la bomba de carga). No fume en el vehículo ni a menos de 7.5 metros. Conecte a tierra el carotante antes de transferir el producto a o desde el contenedor. Cierre y asegure manholes y válvulas, y verifique que éstas no tengan fugas.		
Clasificación de peligro según el Libro Naranja de la ONU: 3 - Líquido inflamable. (Ver decreto 1609 de 2002)		
INFORMACIÓN TOXICOLÓGICA:		
Inhalación: Bajo peligro a temperaturas normales (hasta 38°C). Concentraciones altas del vapor producen irritación de ojos, nariz, garganta y pulmones. Puede afectar al sistema nervioso central ocasionando síntomas como excitación, euforia, dolor de cabeza, mareos, adormecimiento, visión borrosa, fatiga, temblor, convulsiones, pérdida de la conciencia, coma, fallo respiratorio y muerte.		
Contacto con la piel: Baja toxicidad. Al principio pueden no aparecer síntomas; después de algunas horas se presenta inflamación del tejido, decoloración y dolor agudo. Contacto extenso o prolongado puede ocasionar absorción, con síntomas similares a los de inhalación.		
Contacto con los ojos: Exposición a vapores o neblinas produce irritación leve, pero no daños a los tejidos del ojo.		
Ingestión: Baja toxicidad. Puede causar disturbios gastrointestinales, con síntomas como irritación, náusea, vómito y diarrea. Puede afectar el sistema nervioso central, presentándose los síntomas descritos en inhalación. Riesgo de aspiración a los pulmones durante la ingestión o el vómito, lo cual puede producir efectos severos como neumonitis, edema pulmonar o bronconeumonía.		
Efectos crónicos: Piel: efecto desengrasante, enrojecimiento, comezón, inflamación, resquebrajamiento y posible infección secundaria. Reacciones alérgicas en algunos individuos. Carcinogenicidad: Materiales parecidos han producido cáncer en la piel de animales experimentales. La IARC clasifica como posible carcinógeno humano al diesel arrojado por el exhausto de los automotores. La ACGIH clasifica como carcinógeno comprobado para los humanos (clase A1) a las neblinas de aceite mineral medianamente refinado.		
INFORMACIÓN ECOLÓGICA:		
Tóxico para la vida acuática. No permita su entrada a desagües, ríos y otras fuentes de agua.		
Disposición: En caso de seleccionar incineración, verifique que se realice de manera ecológica.		
BIBLIOGRAFÍA:		
1. NFPA. Fire protection guide to Hazardous Materials, 13a. Edición, 2001. Páginas 325M-36, 325M-55. 2. Forsberg, K., et al. Quick selection guide to chemical protective clothing. 3a edición. Van Nostrand Reinhold, 1997. p. 67 3. Naciones Unidas. Recomendaciones para Transporte de Mercancías Peligrosas. USA. 2001. 4. CCOHS. Base de datos MSDS (03-3) en CD-ROM. Canadá. Septiembre de 2003. Registros 1201117, 542527, 503831, 421419. 5. ACGIH. Threshold Limit Values for Chemical Substances (TLVs). USA. 2003. Página 28. 6. Diccionario de química y de productos químicos. Gessner G. Hawley. Ediciones Omega S. A. 1992. Barcelona, p. 292.		
FECHA DE EMISIÓN: Noviembre de 2003		

ACPM

Anexo F. Programación y configuración del microcontrolador

```
;XTAL DE 11.059MHZ BAUDRATE 9.600 SI TH1=#0FDH SMOD=0
$NOMOD51          ; disable predefined 8051 registers
$INCLUDE (REG52.INC) ; include CPU definition file (for example, 8052)
```

```
*****
;
;DATOS ENTRADA Y SALIDA
*****
;
DATO EQU P1
EN EQU P3.2
RS EQU P3.3
ADCK EQU P2
*****
;
;CONTROL ADC
*****
;
SELA EQU P3.4
SELB EQU P3.5
SC EQU P3.6
EOC EQU P3.7
*****
;
*****
;
Bandera EQU 20H
APUNTA EQU 21H
PTX EQU 23H
*****
;
SEN1 EQU 06H
SEN2 EQU 07H
SEN3 EQU 08H
VN EQU 22H
*****
;
;VARIABLES CONVERSION BCD
*****
;
SACTUAL EQU 12H
CEN equ 14H
DECEN equ 15H
UNI equ 16H
*****
;

*****
;
;~~~~~
;~~~~~
;
;VECTORES INICIALIZACION
```



```

;~~~~~
;
;   ORG 00H
;   LJMP MICRO
;   ORG 0BH
;   LJMP INTER
;   ORG 23H
;   LJMP SINTER
;~~~~~
;~~~~~
;
;   ORG 50H
TANQUE: DB'TANQUE#1:  Cm',00H
REF:  DB'REF:150 Cm',00H
;~~~~~
;PROCESO CONFIGURACION INTERUPCIONES
;~~~~~
MICRO:
    MOV SP,#70H      ;Apuntador de Pila
    MOV TMOD,#21H    ;Trabajo el timer 1 en modo 2 PARA LA TRANSMISION
SERIAL
    SETB IP.1

    MOV SCON,#50H
    MOV TH1,#0FDH
    MOV R0,#0BH
    MOV R1,#06H
    MOV R2,#19H
    MOV APUNTA,#00H
    LCALL Ini_LCD
    MOV DPTR,#TANQUE
    LCALL MSG
    LCALL LINEA2
    MOV DPTR,#REF
    LCALL MSG

    MOV IE,#092H     ;Habilita Timer0 y las interrupciones de el puerto serie
    SETB TR0         ;Poner a correr el timer0
    SETB TR1         ;Poner a correr el timer1

AQUI: JNB Bandera.0,AQUI      ; ESPERA 40 milisegundos
    CLR Bandera.0
    LCALL Ini_ADC
    LCALL VALORNIVEL
    LJMP AQUI
;~~~~~
;INTERUPCION

```

```

;~~~~~
INTER:
    MOV TH0,#0FCH
    MOV TL0,#18H
    SETB Bandera.0
RETI

SINTER:
    CLR SCON.0    ;Borra la bandera de RECEPCION R1
RETI
;*****
;*****
;RETARDO LCD
;~~~~~
Esperar_LCD:    ;Retardo de 3 ms

    MOV R3,#25H
Ret1: MOV R4,#19H
Ret2: NOP
    DJNZ R4,Ret2
    DJNZ R3,Ret1
    RET
;*****
;*****
;RETARDO ADC
;*****
RETADC:    ;Retardo de 0.5 ms
    MOV R3,#0A5H
Ret3: NOP
    DJNZ R3,Ret3
    RET
;*****
;*****
;*****VALOR NIVEL *****
VALORNIVEL:
    MOV A,SEN1
    ADD A,SEN2
    ADD A,SEN3
    MOV B,#03H
    DIV AB
    MOV VN,A
    RET
;*****
;*****
;*****

```



```

*****
;
*****
;
;INICIALIZA ADC, GUARDA DATOS, ENVIA DATOS
*****
;
Ini_ADC:
    LCALL RETADC
    SETB SC
    LCALL RETADC
    CLR SC ;Flanco SC
FConv:    JNB EOC,FConv ; ESPERAR Fin de conversion

    MOV @R1,ADCK
IN0:    CJNE R1,#06H,IN1
    MOV SBUF,#01H
    LCALL Esperar_LCD
    MOV SBUF,06H
    CLR SELA
    CLR SELB
IN1:    CJNE R1,#07H,IN2
    MOV SBUF,#02H
    LCALL Esperar_LCD
    MOV A,07H
    ADD A,#01H
    MOV SBUF,A
    SETB SELA
    CLR SELB
IN2:    CJNE R1,#08H,IN3
    MOV SBUF,#03H
    LCALL Esperar_LCD
    MOV A,08H
    ADD A,#02H
    MOV SBUF,A
    CLR SELA
    SETB SELB
IN3:
    INC R1
    CJNE R1,#09H,Ini_ADC
    MOV R1,#06h
    RET
*****
END

```

Anexo G. Código fuente de la configuración inicial

Formulario Configuración inicial

```
Public Dato As Database
Public Tabla As TableDef
Public campo As Field
Public db As Recordset
Dim ws As Workspace
Dim tamaño_archivo As Long
Dim Rtamaño_archivo As Long
Dim rutadb As String

Private Sub Form_Load()
    Configuracion.Visible = True
    On Error GoTo Error
    Creartabla
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub

Private Sub Option1_Click()
    rutadb = App.Path & "/Historial.mdb"
    Set Dato = OpenDatabase(rutadb)
    Set db = Dato.OpenRecordset _
("select * from Central#", dbOpenDynaset, dbOptimistic)
    If db.RecordCount = 0 Then
        db.AddNew
        db!Central1 = "si"
        db!Central2 = "no"
        db!Central3 = "no"
        db.Update
    End If
    If db.RecordCount >= 1 Then
        db.MoveFirst
        db.Edit
        db!Central1 = "si"
        db!Central2 = "no"
        db!Central3 = "no"
        db.Update
    End If
    db.Close
```

```
Dato.Close  
End Sub
```

```
Private Sub Option2_Click()  
rutadb = App.Path & "/Historial.mdb"  
Set Dato = OpenDatabase(rutadb)  
Set db = Dato.OpenRecordset _  
("select * from Central#", dbOpenDynaset, dbOptimistic)
```

```
If db.RecordCount = 0 Then  
db.AddNew  
db!Central1 = "no"  
db!Central2 = "si"  
db!Central3 = "no"  
db.Update  
End If  
If db.RecordCount >= 1 Then  
db.MoveFirst  
db.Edit  
db!Central1 = "no"  
db!Central2 = "si"  
db!Central3 = "no"  
db.Update  
End If  
db.Close  
Dato.Close  
End Sub
```

```
Private Sub Option3_Click()  
rutadb = App.Path & "/Historial.mdb"  
Set Dato = OpenDatabase(rutadb)  
Set db = Dato.OpenRecordset _  
("select * from Central#", dbOpenDynaset, dbOptimistic)
```

```
If db.RecordCount = 0 Then  
db.AddNew  
db!Central1 = "no"  
db!Central2 = "no"  
db!Central3 = "si"  
db.Update  
End If  
If db.RecordCount >= 1 Then  
db.MoveFirst  
db.Edit  
db!Central1 = "no"
```

```

db!Central2 = "no"
db!Central3 = "si"
db.Update
End If
db.Close
Dato.Close
End Sub

```

```

Private Sub Siguiente_Click()
'Configuracion.Visible = False
Direccionip.Visible = True
Direccionip.Aceptar.Visible = False
Direccionip.Cancelar.Visible = False
Direccionip.Command1.Visible = True
Direccionip.Command2.Visible = True
Unload Configuracion
End Sub
Private Sub Command1_Click()
Eliminar
End
End Sub
Sub Creartabla()

```

```

rutadb = App.Path & "/Historial.mdb"
If Len(Dir(rutadb)) <= 0 Then
Set Dato = CreateDatabase(rutadb, dbLangGeneral)
'-----
Set Tabla = Dato.CreateTableDef("Central#")
Set campo = Tabla.CreateField("Central1", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("Central2", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("Central3", dbText, 20)
Tabla.Fields.Append campo
Dato.TableDefs.Append Tabla
'-----
'Direcciones ip
Set Tabla = Dato.CreateTableDef("Direccionesip")
Set campo = Tabla.CreateField("C1ip", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C2ip", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C3ip", dbText, 20)
Tabla.Fields.Append campo
Dato.TableDefs.Append Tabla

```

```

'-----
'Niveles de referencia
Set Tabla = Dato.CreateTableDef("Nivelesref")
Set campo = Tabla.CreateField("C1T1ref", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C1T2ref", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C1T3ref", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C2T1ref", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C2T2ref", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C2T3ref", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C3T1ref", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C3T2ref", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C3T3ref", dbText, 20)
Tabla.Fields.Append campo
Dato.TableDefs.Append Tabla
'-----
'Niveles Registrados
Set Tabla = Dato.CreateTableDef("Registro")
Set campo = Tabla.CreateField("C1T1Nivel", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C1T2Nivel", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C1T3Nivel", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C2T1Nivel", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C2T2Nivel", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C2T3Nivel", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C3T1Nivel", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C3T2Nivel", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("C3T3Nivel", dbText, 20)
Tabla.Fields.Append campo
'-----
Set campo = Tabla.CreateField("Fecha", dbText, 20)

```

```

Tabla.Fields.Append campo
Set campo = Tabla.CreateField("Hora", dbText, 20)
Tabla.Fields.Append campo
Set campo = Tabla.CreateField("Alarma", dbText, 20)
Tabla.Fields.Append campo
'-----
Dato.TableDefs.Append Tabla
Dato.Close
Option1.Value = True
Else
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Central#", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
db.Close
Dato.Close
Configuracion.Visible = False
Principal.Visible = True
End If
End If
End Sub

Private Sub Eliminar()
rutadb = App.Path & "/Historial.mdb"
Kill rutadb
End Sub

```

Anexo H. Código fuente de la direcciones IP

Formulario Direcciones IP

```

Private Sub Aceptar_Click()
If Dir1.Text = "" And Dir2.Text = "" And Dir3.Text = "" Then
Exit Sub
Else
Guardar
End If
Principal.Timer1.Enabled = True
End Sub

Private Sub Cancelar_Click()
Principal.Enabled = True
Unload Direccionip
End Sub
Private Sub Modificar_Click()

```

```

Dir1.Enabled = True
Dir2.Enabled = True
Dir3.Enabled = True
Modificar.Enabled = False
End Sub

```

```

Private Sub Command1_Click()
If Dir1.Text = "" And Dir2.Text = "" And Dir3.Text = "" Then
Dir1.Text = " 0.0.0.0"
Dir2.Text = " 0.0.0.0"
Dir3.Text = " 0.0.0.0"
Guardar
Else
Guardar
End If
Principal.Visible = True
Unload Direccionip
End Sub

```

```

Private Sub Command2_Click()
'Direccionip.Visible = False
Eliminar
'Configuracion.Visible = True
Load Configuracion
Unload Direccionip
End Sub

```

```

Private Sub Form_Load()
'Principal.Visible = False
Verificar
End Sub
Private Sub Form_Unload(Cancel As Integer)
'Principal.Enabled = True
End Sub

```

```

Private Sub Guardar()
'Direccioip.Visible = False
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Direccionesip", dbOpenDynaset, dbOptimistic)
'db.Edit
If db.RecordCount = 0 Then
'db.MoveFirst
db.AddNew

```

```

db!C1ip = Dir1.Text
db!C2ip = Dir2.Text
db!C3ip = Dir3.Text
db.Update
End If
If db.RecordCount >= 1 Then
db.MoveFirst
db.Edit
db!C1ip = Dir1.Text
db!C2ip = Dir2.Text
db!C3ip = Dir3.Text
db.Update
End If
db.Close
Dato.Close
Unload Direccionip
End Sub

```

```

Private Sub Verificar()
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Direccionesip", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Modificar.Visible = True
Dir1.Enabled = False
Dir2.Enabled = False
Dir3.Enabled = False
Dir1.Text = db!C1ip
Dir2.Text = db!C2ip
Dir3.Text = db!C3ip
db.Close
Dato.Close
Else
db.Close
Dato.Close
End If
End Sub

```

```

Private Sub Eliminar()
rutadb = App.Path & "/Historial.mdb"
Kill rutadb
End Sub

```


Anexo I. Código fuente de la niveles de referencia

Formulario Nreferencia

```
Private Sub C1ref_Click()  
REFC1.Visible = True  
End Sub
```

```
Private Sub C2ref_Click()  
REFC2.Visible = True  
End Sub
```

```
Private Sub C3ref_Click()  
REFC3.Visible = True  
End Sub
```

```
Private Sub Form_Load()  
Principal.Enabled = False  
Principal.Timer1.Enabled = False  
End Sub  
Private Sub Form_Unload(Cancel As Integer)  
Principal.Enabled = True  
Principal.Timer1.Enabled = True  
End Sub
```

```
Private Sub Saliref_Click()  
Principal.Enabled = True  
Unload Nreferencia  
End Sub
```

Formulario REFC1

```
Private Sub Aceptar1_Click()  
If Text1.Text = "" And Text2.Text = "" And Text3.Text = "" Then  
Text1.Text = "0"  
Text2.Text = "0"  
Text3.Text = "0"  
Guardar  
Else  
Guardar  
End If  
Unload REFC1  
End Sub
```

```

Private Sub Cancelar1_Click()
Unload REFC1
End Sub
Private Sub Modificar_Click()
Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Modificar.Enabled = False
End Sub
Private Sub Form_Load()
Nreferencia.Enabled = False
Verificar
End Sub
Private Sub Form_Unload(Cancel As Integer)
Nreferencia.Enabled = True
End Sub
Private Sub Verificar()
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Modificar.Visible = True
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
If db!C1T1ref <> "" And db!C1T2ref <> "" And db!C1T3ref <> "" Then
Text1.Text = db!C1T1ref
Text2.Text = db!C1T2ref
Text3.Text = db!C1T3ref
End If
db.Close
Dato.Close
Else
db.Close
Dato.Close
End If
End Sub
Private Sub Guardar()
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)

```

```

If db.RecordCount = 0 Then
db.AddNew
db!C1T1ref = Text1.Text
db!C1T2ref = Text2.Text
db!C1T3ref = Text3.Text
db.Update
End If
If db.RecordCount >= 1 Then
db.MoveFirst
db.Edit
db!C1T1ref = Text1.Text
db!C1T2ref = Text2.Text
db!C1T3ref = Text3.Text
db.Update
End If
db.Close
Dato.Close
End Sub

```

Formulario REFC2

```

Private Sub Aceptar2_Click()
If Text1.Text = "" And Text2.Text = "" And Text3.Text = "" Then
Text1.Text = "0"
Text2.Text = "0"
Text3.Text = "0"
Guardar
Else
Guardar
End If
Unload REFC2
End Sub

```

```

Private Sub Cancelar2_Click()
Unload REFC2
End Sub
Private Sub Modificar_Click()
Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Modificar.Enabled = False

```

```

End Sub

Private Sub Form_Load()
Nreferencia.Enabled = False
Verificar
End Sub

Private Sub Form_Unload(Cancel As Integer)
Nreferencia.Enabled = True
End Sub

Private Sub Verificar()
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Modificar.Visible = True
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
If db!C2T1ref <> "" And db!C2T2ref <> "" And db!C2T3ref <> "" Then
Text1.Text = db!C2T1ref
Text2.Text = db!C2T2ref
Text3.Text = db!C2T3ref
End If
db.Close
Dato.Close
Else
db.Close
Dato.Close
End If
End Sub

Private Sub Guardar()
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)

If db.RecordCount = 0 Then
db.AddNew
db!C2T1ref = Text1.Text
db!C2T2ref = Text2.Text
db!C2T3ref = Text3.Text
db.Update
End If
If db.RecordCount >= 1 Then

```

```

db.MoveFirst
db.Edit
db!C2T1ref = Text1.Text
db!C2T2ref = Text2.Text
db!C2T3ref = Text3.Text
db.Update
End If
db.Close
Dato.Close
End Sub

```

Formulario REFC3

```

Private Sub Aceptar3_Click()
If Text1.Text = "" And Text2.Text = "" And Text3.Text = "" Then
Text1.Text = "0"
Text2.Text = "0"
Text3.Text = "0"
Guardar
Else
Guardar
End If
Unload REFC3
End Sub
Private Sub Cancelar3_Click()
Unload REFC3
End Sub
Private Sub Modificar_Click()
Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Modificar.Enabled = False
End Sub
Private Sub Form_Load()
Nreferencia.Enabled = False
Verificar
End Sub
Private Sub Form_Unload(Cancel As Integer)
Nreferencia.Enabled = True
End Sub
Private Sub Verificar()
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then

```

```

Modificar.Visible = True
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
If db!C3T1ref <> "" And db!C3T2ref <> "" And db!C3T3ref <> "" Then
Text1.Text = db!C3T1ref
Text2.Text = db!C3T2ref
Text3.Text = db!C3T3ref
End If
db.Close
Dato.Close
Else
db.Close
Dato.Close
End If
End Sub
Private Sub Guardar()
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount = 0 Then
db.AddNew
db!C3T1ref = Text1.Text
db!C3T2ref = Text2.Text
db!C3T3ref = Text3.Text
db.Update
End If
If db.RecordCount >= 1 Then
db.MoveFirst
db.Edit
db!C3T1ref = Text1.Text
db!C3T2ref = Text2.Text
db!C3T3ref = Text3.Text
db.Update
End If
db.Close
Dato.Close
End Sub

```

Anexo J. Código fuente principal

Formulario Principal

```
Public contenido_archivo As String, nombre_archivo As String, ruta_remota As String
```

```
Public ruta As String, archivo_temporal As String
```

```
Public Dato As Database
```

```
Public Tabla As TableDef
```

```
Public campo As Field
```

```
Public db As Recordset
```

```
Dim ws As Workspace
```

```
Dim tamaño_archivo As Long
```

```
Dim Rtamaño_archivo As Long
```

```
Dim rutadb As String
```

```
*****
```

```
'Datos Serial
```

```
*****
```

```
Dim DataIN, DataOUT As String
```

```
Dim FR, FT, CIN As Integer
```

```
Public NTANQUE1, NTANQUE2, NTANQUE3 As Integer
```

```
Dim Tip1, Tip2, Tip3, Tip4 As String
```

```
Dim i As Integer
```

```
*****
```

```
*****
```

```
'Datos Cliente_servidor
```

```
*****
```

```
Public NC1T1, NC1T2, NC1T3 As String
```

```
Public NC2T1, NC2T2, NC2T3 As String
```

```
Public NC3T1, NC3T2, NC3T3 As String
```

```
*****
```

```
Public Dir1, Dir2, Dir3 As String
```

```
Public Cen1, Cen2, Cen3 As String
```

```
*****
```

```
Private Sub Form_Load()
```

```
*****
```

```
Configuracion Serial
```

```
*****
```

```
MSComm.Settings = "9600,N,8,1" ' velocidad, paridad
```

```
MSComm.CommPort = 1 'elegir el #del puerto
```

```
MSComm.PortOpen = True 'abrir el puerto
```

```
MSComm.RThreshold = 1 'Debe ser mayor a 0
```

```
MSComm.Handshaking = comNone
```

```
MSComm.InputLen = 1
```

```
*****
```

```
Central
```

```
DirIP
```

```
If Cen1 = "si" Then
```

```
Servidor
```

```
End If
```

```
If Cen1 = "no" Then
```

```
Cliente
```

```
End If
```

```
End Sub
```

```
Private Sub ws_server_DataArrival(Index As Integer, ByVal bytesTotal As Long)
```

```
Dim datos As String
```

```
*****
```

```
*****
```

```
For i = 0 To ws_server.UBound 'ns - 1
```

```
If ws_server(i).State = sckConnected Then
```

```
ws_server(i).GetData datos
```

```
If ws_server(i).RemoteHostIP = Dir1 Then
```

```
If Right(datos, 4) = "C1T1" Then
```

```
NC1T1 = Left(datos, (Len(datos) - 4))
```

```
End If
```

```
If Right(datos, 4) = "C1T2" Then
```

```
NC1T2 = Left(datos, (Len(datos) - 4))
```

```
End If
```

```
If Right(datos, 4) = "C1T3" Then
```

```
NC1T3 = Left(datos, (Len(datos) - 4))
```

```
End If
```

```
End If
```

```
If ws_server(i).RemoteHostIP = Dir2 Then "172.16.49.183" Then
```

```
If Right(datos, 4) = "C2T1" Then
```

```
NC2T1 = Left(datos, (Len(datos) - 4))
```

```
End If
```

```
If Right(datos, 4) = "C2T2" Then
```

```
NC2T2 = Left(datos, (Len(datos) - 4))
```

```
End If
```

```
If Right(datos, 4) = "C2T3" Then
```

```
NC2T3 = Left(datos, (Len(datos) - 4))
```

```
End If
```

```
End If
```



```

If ws_server(i).RemoteHostIP = Dir3 Then "172.16.49.154" Then
If Right(datos, 4) = "C3T1" Then
    NC3T1 = Left(datos, (Len(datos) - 4))
End If
If Right(datos, 4) = "C3T2" Then
    NC3T2 = Left(datos, (Len(datos) - 4))
End If
If Right(datos, 4) = "C3T3" Then
    NC3T3 = Left(datos, (Len(datos) - 4))
End If
End If
End If
Next

End Sub
Private Sub ws_cliente_DataArrival(Index As Integer, ByVal bytesTotal As Long)
Dim datosR As String
Dim j As Integer
j = j + 1
ws_cliente(0).GetData datosR
If ws_cliente(0).State = sckConnected Then
If Cen2 = "si" Then
If Right(datosR, 4) = "C1T1" Then
    NC1T1 = Left(datosR, (Len(datosR) - 4))
End If
If Right(datosR, 4) = "C1T2" Then
    NC1T2 = Left(datosR, (Len(datosR) - 4))
End If
If Right(datosR, 4) = "C1T3" Then
    NC1T3 = Left(datosR, (Len(datosR) - 4))
End If
If Right(datosR, 4) = "C3T1" Then
    NC3T1 = Left(datosR, (Len(datosR) - 4))
End If
If Right(datosR, 4) = "C3T2" Then
    NC3T2 = Left(datosR, (Len(datosR) - 4))
End If
If Right(datosR, 4) = "C3T3" Then
    NC3T3 = Left(datosR, (Len(datosR) - 4))
End If
End If
*****
If Cen3 = "si" Then
If Right(datosR, 4) = "C1T1" Then
    NC1T1 = Left(datosR, (Len(datosR) - 4))

```

```

End If
If Right(datosR, 4) = "C1T2" Then
    NC1T2 = Left(datosR, (Len(datosR) - 4))
End If
If Right(datosR, 4) = "C1T3" Then
    NC1T3 = Left(datosR, (Len(datosR) - 4))
End If
If Right(datosR, 4) = "C2T1" Then
    NC2T1 = Left(datosR, (Len(datosR) - 4))
End If
If Right(datosR, 4) = "C2T2" Then
    NC2T2 = Left(datosR, (Len(datosR) - 4))
End If
If Right(datosR, 4) = "C2T3" Then
    NC2T3 = Left(datosR, (Len(datosR) - 4))
End If
End If
End If
End Sub
Private Sub r_server_ConnectionRequest(ByVal requestID As Long)
    ns = CargarSocket
    ws_server(ns).Accept requestID
End Sub
Private Sub MSComm_OnComm()
    *****
    Select Case MSComm.CommEvent
    ' Eventos Posibles
    Case comEvReceive
    If MSComm.Input = "" Then
        Exit Sub
    End If
    DataIN = Asc(MSComm.Input)
    CIN = CIN + 1

    *****

    If CIN = 1 Then 'And FR = 1 Then
        'Text1.Text = DataIN
        If Cen1 = "si" Then
            NC1T1 = Val(DataIN)
        End If
        If Cen2 = "si" Then
            NC2T1 = Val(DataIN)
        End If
        If Cen3 = "si" Then

```

```

NC3T1 = Val(DataIN)
End If
'CIN = 0
End If
If CIN = 2 Then 'And FR = 2 Then
'Text2.Text = DataIN
If Cen1 = "si" Then
NC1T2 = Val(DataIN)
End If
If Cen2 = "si" Then
NC2T2 = Val(DataIN)
End If
If Cen3 = "si" Then
NC3T2 = Val(DataIN)
End If
'CIN = 0
End If
If CIN = 3 Then 'And FR = 3 Then
'Text3.Text = DataIN
If Cen1 = "si" Then
NC1T3 = Val(DataIN)
End If
If Cen2 = "si" Then
NC2T3 = Val(DataIN)
End If
If Cen3 = "si" Then
NC3T3 = Val(DataIN)
End If
CIN = 0
End If
End Select
*****

DatosRnd
End Sub
Private Sub C1_Click()
Central1.Visible = True
End Sub

Private Sub C2_Click()
Central2.Visible = True
End Sub

Private Sub C3_Click()
Central3.Visible = True
End Sub

```

```

Private Sub Salir_Click()
*****

If MSComm.PortOpen = True Then
MSComm.PortOpen = False 'cerrar el puerto
End If
*****

If Cen1 = "no" Then
Me.ws_cliente(0).Close
End If
If Cen1 = "si" Then
For i = 0 To ns - 1
Me.ws_server(i).Close
Next
End If
*****

End
End Sub
Private Sub ws_cliente_Close(Index As Integer)
'cierra la conexion
Estado(0).Caption = "Desconectado"
ws_cliente(0).Close
End Sub
Private Sub ws_server_Close(Index As Integer)
'cierra la conexion
For i = 0 To ws_server.UBound 'ns - 1
If ws_server(i).State = sckClosing Then
ws_server(i).Close
End If
Next
End Sub

Private Function CargarSocket() As Integer
Dim nsockets As Integer 'numero de sockets
Dim x As Integer 'contador

nsockets = ws_server.UBound
For x = 0 To nsockets
If ws_server(x).State = sckClosed Then
CargarSocket = x 'retorna el indice
Exit Function 'abandona la funcion
End If
Next
Load ws_server(nsockets + 1)
'devuelve el nuevo indice

```

```

CargarSocket = ws_server.UBound
End Function
Private Sub Cliente()
For i = 0 To ws_server.UBound 'ns - 1
Me.ws_server(i).Close
Next
End Sub
Private Sub Servidor()
Me.ws_cliente(0).Close
*****

Me.r_server.Close
'asignamos un puerto
Me.r_server.LocalPort = "8080"
'ponemos a la escucha el puerto asignado
Me.r_server.Listen
*****

End Sub
Private Sub Conectar()
ws_cliente(0).Close
ws_cliente(0).RemoteHost = Dir1
ws_cliente(0).RemotePort = "8080"
ws_cliente(0).Connect
End Sub

Private Sub ws_cliente_Connect(Index As Integer)
Estado(0).Caption = "Conectado"
End Sub

Private Sub Central()
On Error GoTo Error
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Central#", dbOpenDynaset, dbOptimistic)
If db!Central1 = "si" Then
Cen1 = "si"
Cen2 = "no"
Cen3 = "no"
C1.Top = 1080
C1.Left = 1920
End If
If db!Central2 = "si" Then
Cen2 = "si"
Cen1 = "no"
Cen13 = "no"

```

```

C2.Top = 1080
C2.Left = 1920
C1.Top = 2640
C1.Left = 360
End If
If db!Central3 = "si" Then
Cen3 = "si"
Cen1 = "no"
Cen3 = "no"
C3.Top = 1080
C3.Left = 1920
C1.Top = 2640
C1.Left = 3600
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub
Private Sub DirIP()
On Error GoTo Error
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Direccionesip", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
If db!C1ip <> "" And db!C2ip <> "" And db!C3ip <> "" Then
Dir1 = db!C1ip
Dir2 = db!C2ip
Dir3 = db!C3ip
End If
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub
Private Sub Guardar()
On Error GoTo Error

```

```

rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Registro", dbOpenDynaset, dbOptimistic)
db.MoveLast
db.AddNew
db!C1T1Nivel = NC1T1
db!C1T2Nivel = NC1T2
db!C1T3Nivel = NC1T3
db!C2T1Nivel = NC2T1
db!C2T2Nivel = NC2T2
db!C2T3Nivel = NC2T3
db!C3T1Nivel = NC3T1
db!C3T2Nivel = NC3T2
db!C3T3Nivel = NC3T3
*****

db!Fecha = Date
db!Hora = Time
*****

db.Update
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub
Private Sub DatosRnd()
*****

If Cen1 = "si" Then
NC1T1 = Int(180) * Rnd
NC1T2 = Int(180) * Rnd
NC1T3 = Int(180) * Rnd
NC3T1 = Int(180) * Rnd
NC3T2 = Int(180) * Rnd
NC3T3 = Int(180) * Rnd
End If
*****

If Cen2 = "si" Then
NC2T1 = Int(180) * Rnd
NC2T2 = Int(180) * Rnd
NC2T3 = Int(180) * Rnd
End If
*****

```

```

If Cen3 = "si" Then
NC3T1 = Int(180) * Rnd
NC3T2 = Int(180) * Rnd
NC3T3 = Int(180) * Rnd
End If
End Sub
Private Sub Enviar()
Static k As Integer

'*****

If Cen1 = "si" Then
For i = 0 To ws_server.UBound 'ns - 1
If (ws_server(i).State = sckConnected) Then
k = k + 1
If ws_server(i).RemoteHostIP = Dir2 Then
If k = 1 Then
ws_server(i).SendData (NC1T1 & "C1T1")
End If
If k = 2 Then
ws_server(i).SendData (NC1T2 & "C1T2")
End If
If k = 3 Then
ws_server(i).SendData (NC1T3 & "C1T3")
End If
If k = 4 Then
ws_server(i).SendData (NC3T1 & "C3T1")
End If
If k = 5 Then
ws_server(i).SendData (NC3T2 & "C3T2")
End If
If k = 6 Then
ws_server(i).SendData (NC3T3 & "C3T3")
k = 0
End If
End If

'*****

If ws_server(i).RemoteHostIP = Dir3 Then
If k = 1 Then
ws_server(i).SendData (NC1T1 & "C1T1")
End If
If k = 2 Then
ws_server(i).SendData (NC1T2 & "C1T2")
End If
If k = 3 Then
ws_server(i).SendData (NC1T3 & "C1T3")

```



```

End If
If k = 4 Then
ws_server(i).SendData (NC2T1 & "C2T1")
End If
If k = 5 Then
ws_server(i).SendData (NC2T2 & "C2T2")
End If
If k = 6 Then
ws_server(i).SendData (NC2T3 & "C2T3")
k = 0
End If
End If
End If
Next
End If
*****

If Cen2 = "si" Then
If ws_cliente(0).State = sckConnected Then
k = k + 1
If k = 1 Then
ws_cliente(0).SendData (NC2T1 & "C2T1")
End If
If k = 2 Then
ws_cliente(0).SendData (NC2T2 & "C2T2")
End If
If k = 3 Then
ws_cliente(0).SendData (NC2T3 & "C2T3")
k = 0
End If
End If
End If
*****

If Cen3 = "si" Then
If ws_cliente(0).State = sckConnected Then
k = k + 1
If k = 1 Then
ws_cliente(0).SendData (NC3T1 & "C3T1")
End If
If k = 2 Then
ws_cliente(0).SendData (NC3T2 & "C3T2")
End If
If k = 3 Then
ws_cliente(0).SendData (NC3T3 & "C3T3")
k = 0

```

```
End If
End If
End If
```

```
End Sub
Private Sub mnuRef_Click()
Nreferencia.Visible = True
Timer1.Enabled = False
End Sub
Private Sub mnuSalir_Click()
End
End Sub
```

```
Private Sub Opendir_Click()
Direccionip.Visible = True
Timer1.Enabled = False
End Sub
```

```
Private Sub Timer1_Timer()
If Cen1 = "no" Then
If ws_cliente(0).State = sckClosed Then
Conectar
End If
End If
*****
```

```
End Sub
```

```
Private Sub Timer2_Timer()
DatosRnd
```

```
Enviar
```

```
End Sub
Private Sub ws_server_Error(Index As Integer, ByVal Number As Integer,
Description As String, ByVal Scode As Long, ByVal Source As String, ByVal
HelpFile As String, ByVal HelpContext As Long, CancelDisplay As Boolean)
'cerramos la conexion
For i = 0 To ws_server.UBound 'ns - 1
If ws_server(i).State = sckClosing Then
ws_server(i).Close
End If
Next
'mostramos informacion sobre el error
```

```

Respuesta = MsgBox("Intento de conexion fallido", vbOKOnly + vbExclamation +
vbApplicationModal + vbSystemModal, "Conexion Rechazada")
If Respuesta = vbOK Then
Exit Sub
End If
End Sub
Private Sub ws_cliente_Error(Index As Integer, ByVal Number As Integer,
Description As String, ByVal Scode As Long, ByVal Source As String, ByVal
HelpFile As String, ByVal HelpContext As Long, CancelDisplay As Boolean)

'cerramos la conexion

ws_cliente(0).Close
Timer1.Enabled = False
'mostramos informacion sobre el error

'MsgBox "Error numero " & Number & ": " & Description, vbInformation
Respuesta = MsgBox("Central Principal Fuera de Alcance", vbRetryCancel +
vbExclamation + vbApplicationModal + vbSystemModal, "Conexion Rechazada")
If Respuesta = vbRetry Then
Timer1.Enabled = True
Exit Sub
End If
If Respuesta = vbCancel Then
Timer1.Enabled = False
End If
End Sub

```

Anexo K. Código fuente del nivel de cada Tanque

Formulario Central1

```

Dim i As Integer
Const max = 180
Const altura = 1815
Private Sub C1T1_Click()
Central1.Caption = "Central 1 - Tanque 1"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
Label5.Visible = True
Label7.Visible = False
Text1.Visible = True

```

```

Text2.Visible = True
C1T1.Visible = False
C1T2.Visible = False
C1T3.Visible = False
Salir1.Visible = True
Salir.Visible = False
Central1.Width = 6405
Shape1.Visible = True
Line1.Visible = True
Line2.Visible = True
Label4.Visible = True
Label6.Visible = True
On Error GoTo Error
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Text2.Text = db!C1T1ref
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub

Private Sub C1T2_Click()
Central1.Caption = "Central 1 - Tanque 2"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
Label5.Visible = True
Label7.Visible = False
Text1.Visible = True
Text2.Visible = True
C1T1.Visible = False
C1T2.Visible = False
C1T3.Visible = False
Salir1.Visible = True
Salir.Visible = False
Central1.Width = 6405
Shape1.Visible = True

```

```

Line1.Visible = True
Line2.Visible = True
Label4.Visible = True
Label6.Visible = True
On Error GoTo Error
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Text2.Text = db!C1T2ref
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub

```

```

Private Sub C1T3_Click()
Central1.Caption = "Central 1 - Tanque 3"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
Label5.Visible = True
Label7.Visible = False
Text1.Visible = True
Text2.Visible = True
C1T1.Visible = False
C1T2.Visible = False
C1T3.Visible = False
Salir1.Visible = True
Salir.Visible = False
Central1.Width = 6405
Shape1.Visible = True
Line1.Visible = True
Line2.Visible = True
Label4.Visible = True
Label6.Visible = True
On Error GoTo Error
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _

```

```

("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Text2.Text = db!C1T3ref
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub

```

```

Private Sub Form_Load()
Label1.Visible = False
Label2.Visible = False
Label3.Visible = False
Label5.Visible = False
Label7.Caption = Me.Caption
Text1.Visible = False
Text2.Visible = False
Salir1.Visible = False
Text2.Enabled = False
Text1.Enabled = False
Central1.Width = 4545
Shape1.Visible = False
Line1.Visible = False
Line2.Visible = False
Label4.Visible = False
Label6.Visible = False
End Sub

```

```

Private Sub Salir_Click()
Unload Central1
End Sub

```

```

Private Sub Salir1_Click()
Central1.Caption = "Monitoreo Central 1"
Label1.Visible = False
Label2.Visible = False
Label3.Visible = False
Label5.Visible = False
Label7.Visible = True
Text1.Visible = False
Text2.Visible = False

```

```

Salir1.Visible = False
Salir.Visible = True
C1T1.Visible = True
C1T2.Visible = True
C1T3.Visible = True
Central1.Width = 4545
Shape1.Visible = False
Line1.Visible = False
Line2.Visible = False
Label4.Visible = False
Label6.Visible = False
End Sub

```

```

Private Sub Timer1_Timer()
If Val(Text1.Text) < max + 1 Then
i = Val(Text1.Text) * (altura / max)
Shape2.Height = altura - i
End If
End Sub

```

```

Private Sub Timer2_Timer()
If Central1.Caption = "Central 1 - Tanque 1" Then
Text1.Text = Principal.NC1T1
End If
If Central1.Caption = "Central 1 - Tanque 2" Then
Text1.Text = Principal.NC1T2
End If
If Central1.Caption = "Central 1 - Tanque 3" Then
Text1.Text = Principal.NC1T3
End If
End Sub

```

Formulario Central2

```

Dim i As Integer
Const max = 180
Const altura = 1815
Private Sub C2T1_Click()
Central2.Caption = "Central 2 - Tanque 1"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True

```

```

Label5.Visible = True
Label7.Visible = False
Text1.Visible = True
Text2.Visible = True
C2T1.Visible = False
C2T2.Visible = False
C2T3.Visible = False
Salir1.Visible = True
Salir.Visible = False
Central2.Width = 6405
Shape1.Visible = True
Line1.Visible = True
Line2.Visible = True
Label4.Visible = True
Label6.Visible = True
On Error GoTo Error
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Text2.Text = db!C2T1ref
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub

Private Sub C2T2_Click()
Central2.Caption = "Central 2 - Tanque 2"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
Label5.Visible = True
Label7.Visible = False
Text1.Visible = True
Text2.Visible = True
C2T1.Visible = False
C2T2.Visible = False
C2T3.Visible = False
Salir1.Visible = True

```



```

Salir.Visible = False
Central2.Width = 6405
Shape1.Visible = True
Line1.Visible = True
Line2.Visible = True
Label4.Visible = True
Label6.Visible = True
On Error GoTo Error
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Text2.Text = db!C2T2ref
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub

```

```

Private Sub C2T3_Click()
Central2.Caption = "Central 2 - Tanque 3"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
Label5.Visible = True
Label7.Visible = False
Text1.Visible = True
Text2.Visible = True
C2T1.Visible = False
C2T2.Visible = False
C2T3.Visible = False
Salir1.Visible = True
Salir.Visible = False
Central2.Width = 6405
Shape1.Visible = True
Line1.Visible = True
Line2.Visible = True
Label4.Visible = True
Label6.Visible = True
On Error GoTo Error

```

```

rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Text2.Text = db!C2T3ref
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
End
End Sub

```

```

Private Sub Form_Load()
Label1.Visible = False
Label2.Visible = False
Label3.Visible = False
Label5.Visible = False
Label7.Caption = Me.Caption
Text1.Visible = False
Text2.Visible = False
Salir1.Visible = False
Text2.Enabled = False
Text1.Enabled = False
Central2.Width = 4545
Shape1.Visible = False
Line1.Visible = False
Line2.Visible = False
Label4.Visible = False
Label6.Visible = False
End Sub

```

```

Private Sub Salir_Click()
Unload Central2
End Sub

```

```

Private Sub Salir1_Click()
Central2.Caption = "Monitoreo Central 2"
Label1.Visible = False
Label2.Visible = False

```

```

Label3.Visible = False
Label5.Visible = False
Label7.Visible = True
Text1.Visible = False
Text2.Visible = False
Salir1.Visible = False
Salir.Visible = True
C2T1.Visible = True
C2T2.Visible = True
C2T3.Visible = True
Central2.Width = 4545
Shape1.Visible = False
Line1.Visible = False
Line2.Visible = False
Label4.Visible = False
Label6.Visible = False
End Sub

```

```

Private Sub Timer1_Timer()
If Val(Text1.Text) < max + 1 Then
i = Val(Text1.Text) * (altura / max)
Shape2.Height = altura - i
End If
End Sub

```

```

Private Sub Timer2_Timer()
If Central2.Caption = "Central 2 - Tanque 1" Then
Text1.Text = Principal.NC2T1
End If
If Central2.Caption = "Central 2 - Tanque 2" Then
Text1.Text = Principal.NC2T2
End If
If Central2.Caption = "Central 2 - Tanque 3" Then
Text1.Text = Principal.NC2T3
End If
End Sub

```

Formulario Central3

```

Dim i As Integer
Const max = 180
Const altura = 1815
Private Sub C3T1_Click()
Central3.Caption = "Central 3 - Tanque 1"

```

```

Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
Label5.Visible = True
Label7.Visible = False
Text1.Visible = True
Text2.Visible = True
C3T1.Visible = False
C3T2.Visible = False
C3T3.Visible = False
Salir1.Visible = True
Salir.Visible = False
Central3.Width = 6405
Shape1.Visible = True
Line1.Visible = True
Line2.Visible = True
Label4.Visible = True
Label6.Visible = True
On Error GoTo Error
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Text2.Text = db!C3T1ref
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub
Private Sub C3T2_Click()
Central3.Caption = "Central 3 - Tanque 2"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
Label5.Visible = True
Label7.Visible = False
Text1.Visible = True
Text2.Visible = True
C3T1.Visible = False
C3T2.Visible = False

```

```

C3T3.Visible = False
Salir1.Visible = True
Salir.Visible = False
Central3.Width = 6405
Shape1.Visible = True
Line1.Visible = True
Line2.Visible = True
Label4.Visible = True
Label6.Visible = True
On Error GoTo Error
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Text2.Text = db!C3T2ref
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub

```

```

Private Sub C3T3_Click()
Central3.Caption = "Central 3 - Tanque 3"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
Label5.Visible = True
Label7.Visible = False
Text1.Visible = True
Text2.Visible = True
C3T1.Visible = False
C3T2.Visible = False
C3T3.Visible = False
Salir1.Visible = True
Salir.Visible = False
Central3.Width = 6405
Shape1.Visible = True
Line1.Visible = True
Line2.Visible = True
Label4.Visible = True

```

```

Label6.Visible = True
On Error GoTo Error
rutadb = App.Path & "/Historial.mdb"
Set Dato = OpenDatabase(rutadb)
Set db = Dato.OpenRecordset _
("select * from Nivelesref", dbOpenDynaset, dbOptimistic)
If db.RecordCount >= 1 Then
Text2.Text = db!C3T3ref
End If
db.Close
Dato.Close
Exit Sub
Error:
    MsgBox Err.Description
    Err.Clear
    End
End Sub

```

```

Private Sub Form_Load()
Label1.Visible = False
Label2.Visible = False
Label3.Visible = False
Label5.Visible = False
Label7.Caption = Me.Caption
Text1.Visible = False
Text2.Visible = False
Salir1.Visible = False
Text2.Enabled = False
Text1.Enabled = False
Central3.Width = 4545
Shape1.Visible = False
Line1.Visible = False
Line2.Visible = False
Label4.Visible = False
Label6.Visible = False
End Sub

```

```

Private Sub Salir_Click()
Unload Central3
End Sub

```

```

Private Sub Salir1_Click()
Central3.Caption = "Monitoreo Central 3"
Label1.Visible = False
Label2.Visible = False

```

```

Label3.Visible = False
Label5.Visible = False
Label7.Visible = True
Text1.Visible = False
Text2.Visible = False
Salir1.Visible = False
Salir.Visible = True
C3T1.Visible = True
C3T2.Visible = True
C3T3.Visible = True
Central3.Width = 4545
Shape1.Visible = False
Line1.Visible = False
Line2.Visible = False
Label4.Visible = False
Label6.Visible = False
End Sub

```

```

Private Sub Timer1_Timer()
If Val(Text1.Text) < max + 1 Then
i = Val(Text1.Text) * (altura / max)
Shape2.Height = altura - i
End If
End Sub

```

```

Private Sub Timer2_Timer()
If Central3.Caption = "Central 3 - Tanque 1" Then
Text1.Text = Principal.NC3T1
End If
If Central3.Caption = "Central 3 - Tanque 2" Then
Text1.Text = Principal.NC3T2
End If
If Central3.Caption = "Central 3 - Tanque 3" Then
Text1.Text = Principal.NC3T3
End If
End Sub

```